Fast Search Algorithms for IC Printed Mark Quality Inspection

A thesis presented

by

Ming-Ching Chang

to

The Department of Computer Science and Information Engineering

in partial fulfillment of the requirements

for the degree of

Master of Science

in the subject of

Computer Science and Information Engineering

National Taiwan University

Taipei, Taiwan

June 8, 1998

Abstract

This paper presents an effective and general purpose search algorithm for alignment, and we applied it to IC printed mark quality inspection. The search procedure is based on normalized cross correlation, and we improve the method with hierarchical resolution pyramid, dynamic programming, subpixel accuracy, multiple target search, and automatic model selection. The proposed search method can be applied to general visual inspection.

The IC printed mark includes a logo pattern and characters. Due to the alignment error of the inspection machine, the mark can be rotated or translated. Main printing error of an IC mark includes: distortion, missing ink, wrong position, double print, smear print, bad contrast (global or partial character), misprint, and mis-orientation print. The inspection accuracy, speed, reliability, and repeatability are all important for the industrial requirement.

To develop the inspection algorithm, digital image processing and computer vision techniques including image binarization, projection, image difference, normalized cross correlation, and mathematical morphology are used. We develop the teaching and inspection function and optimize the system and test it on an IC inspection machine. Our algorithm achieves high accuracy, reliability, and repeatability with high speed for industrial requirement and works well on field test of various IC products.

To my father, Sheng-Chu; mother, Lan-Teng; and all my families and friends

Acknowledgements

Many people supported me much in writing this thesis. First of all, I would like to thank my advisor, Associate Professor Chiou-Shann Fuh. His aggressive, determined, and earnest attitude had inspired me to work constructively and influenced my point of view of handling things. His positive criticism and comments on the drafts of this document had advanced the quality and content of this thesis.

The days in Intelligent Robotics Laboratory (IRL) were filled with gratification, pleasure, and atmosphere of research. I greatly appreciate the friendship, concern, and words of cheer from classmates in IRL. I would also appreciate my supervisor and all the members of Mechanical Industrial Research Laboratories, Industrial Technology Research Institute (MIRL/ITRI) sincerely. Their assistance and guidance had helped me much and given me a clear future direction.

I would thank my advisors, associates, and friends who support me, discuss my work, share their ideas, and recommend avenues to break through bottlenecks and difficulties. During the incubating and composing periods of this thesis, my parents, elder brother, girl friend, and all of my dear families and friends devoted themselves to support me with their passion, care, and encouragement. I will always cherish these and be proud of all of you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Thesis Motivation

Integrated Circuits (IC) are the fundamentals of computer and electronic industry. IC industry is also the important topic and weapon to enhance our industry and to compete in worldwide market. IC industry includes wafer fabrication process in the front end and chip packaging in the back end.

Mask exposure and defect inspection in wafer fabrication process and IC printed mark inspection, pin inspection, or die bonding in chip packaging all require high precision alignment. High precision alignment enables precise and efficient implementation of defect inspection and die bonding. Due to the rapid increase of pin counts and circuit density, efficient and high accuracy alignment is a required and important algorithm.

IC printed marking is the final stage of chip packaging to print product number and trade mark on the chip to identify product function and classification. IC printed mark is the first and most prominent part a user sees. The user usually associates IC printed mark quality with chip function and quality. In the endless pursuit of quality perfection, enhancing IC printed mark quality is of utmost importance.

IC chips are mass produced. Traditional inspection is manual and relies on eyes.

In industrialized countries, labor is expensive and inspection is monotonous, laborious, fatiguing, and prone to mistakes. Employees are unwilling and it is impossible to do IC printed mark quality control or inspection manually with eyes. Automatic IC printed mark quality inspection with computer vision is a natural and unstoppable trend.

We develop the fast search algorithm and apply to IC printed mark quality inspection. We collect various kinds of searching methods and develop an software solution to match a 128×128 pattern in a 640×480 field of view; the computing time is within 50 to 70 ms, and the locating accuracy can achieve subpixel accuracy.

The IC printed mark includes a logo pattern and characters. Due to the alignment error of the inspection machine, the mark can be rotated or translated. Main printing error of an IC mark includes: distortion, missing ink, wrong position, double print, smear print, bad contrast (global or partial character), misprint, and mis-orientation print as shown in Figures 1.1 and 1.2 [2, 6].

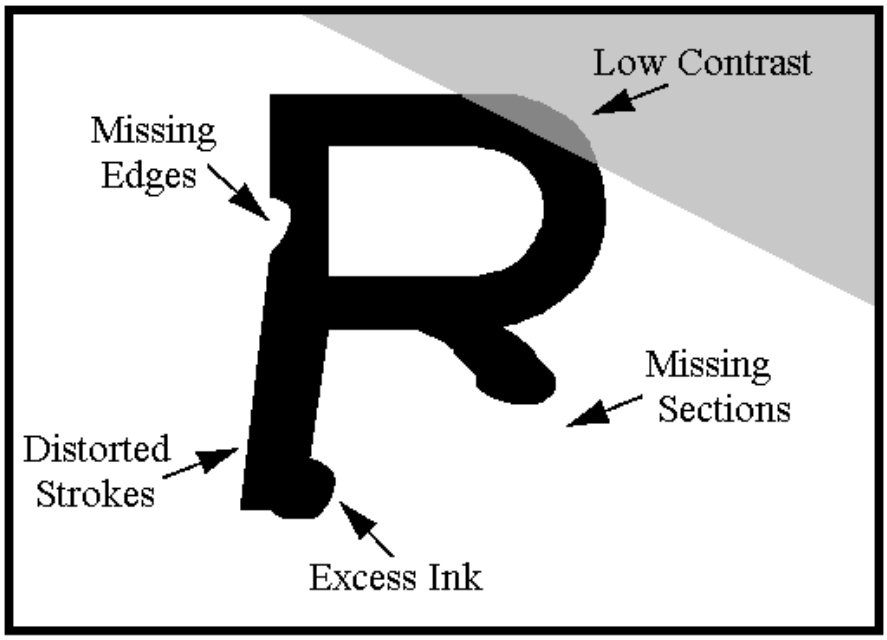

Figure 1.1 : *IC printed mark errors.*

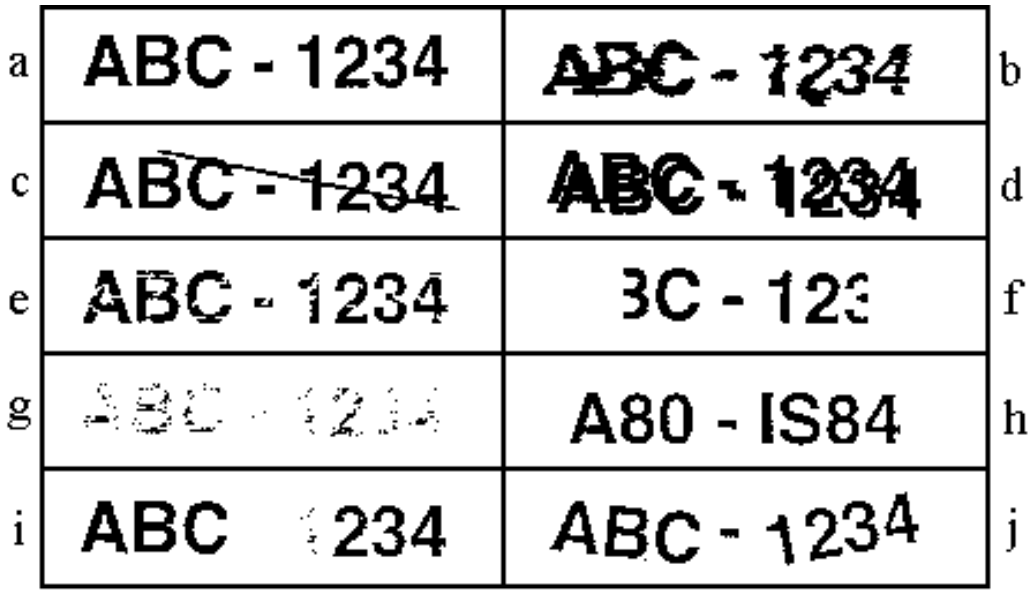

Figure 1.2 : *Main IC printed mark errors: (a) good, (b) smeared, (c) scraped, (d) double print, (e) broken, (f) missing ink, (g) bad contrast, (h) misprinted, (i) partial bad contrast, and (j) mis-orientation.*

1.2 The Proposed Method

We develop the IC printed mark quality inspection algorithm and build a GUI (Graphical User Interface) environment. An operator can teach and adjust a good IC mark sample and inspect a batch of ICs based on the golden sample.

Alignment and search is the most important part of the optical inspection. Typical 2D image transform includes translation, rotation, scaling, tilting, skew, and other 2D nonlinear transform. In 3D perspective vision, image translation, rotation, and scaling often occurs. In real case of industrial inspection, we can assume no image scaling, because the distance between camera and inspected parts is fixed. Our fast search algorithm can solve image translation and rotation in realtime and reach the subpixel matching accuracy.

After accurate alignment, optical verification can be easily achieved by image dif-

ference. Due to the matching error, some edge noise will remain after image difference. Morphological opening will eliminate the edge noise and leave the defect alone. We can verify the inspection result and make decision to accept or reject this part.

Normalized Correlation Search (NCS) is the best linear method to solve 2D image matching [38], but traditional image correlation is computationally inefficient. We use the hierarchical pyramid strategy and a dynamic programming method to improve the speed and keep the excellent reliability and accuracy of NCS. In addition, subpixel search accuracy, multiple target search, and automatic search model detection are added to improve the matching functionality.

We use two fiducial marks to solve the image rotation [7]. By calculating the matching angle of the two fiducial marks, we get the rotation angle. Tolerance of about $\pm 10^{\circ}$ is acceptable for NCS due to its excellent matching ability and reliability.

The inspection method is divided into two main steps: the teaching step and the inspection step. In the teaching step the operator teaches a golden sample to the program. We assume the teaching IC sample is perfect i.e. the printed mark has no defects and no rotation and it is well lighted. After learning many features of the golden image, we perform inspection based on the taught data. During teaching step, we focus on complete functions for teaching on various kinds of IC printed marks. On the other hand, during inspection step, we focus on inspecting speed and accuracy for industrial requirement.

This thesis is organized as follows. Chapter 1 gives introduction, thesis motivation, and our proposed method. Chapter 2 show some theoretical background including projection, image binarization, image difference, mathematical morphology, and normalized cross correlation. Chapter 3 describes the fast search algorithm for inspection alignment. Chapter 4 explains the IC printed mark teaching and inspection process, including discussion about the threshold parameters of the system. Chapter 5 gives some conclusions.

Chapter 2

Background

2.1 Projection

Projection [17] can be used to segment image especially for blocks where bounding boxes separate from each other. Projection can be binary or grayscale with proper threshold. Projection can be horizontal, vertical, or at any direction. Let digital image I have pixels $I(r, c)$, where $1 \leq r \leq R$ and $1 \leq c \leq C$, then
horizontal projection is

$$P_H(r) = \#\{c|(r, c) \in I\} \tag{2.1}$$

vertical projection is

$$P_V(c) = \#\{r|(r, c) \in I\} \tag{2.2}$$

Based on the nature of the IC printed marks, we can use horizontal projection to segment each printed line and vertical projection to segment each printed character as shown in Figure 2.1.

We use binary projection to segment IC printed words and marks, so a proper threshold is needed. the threshold selection is described in Section 2.2.

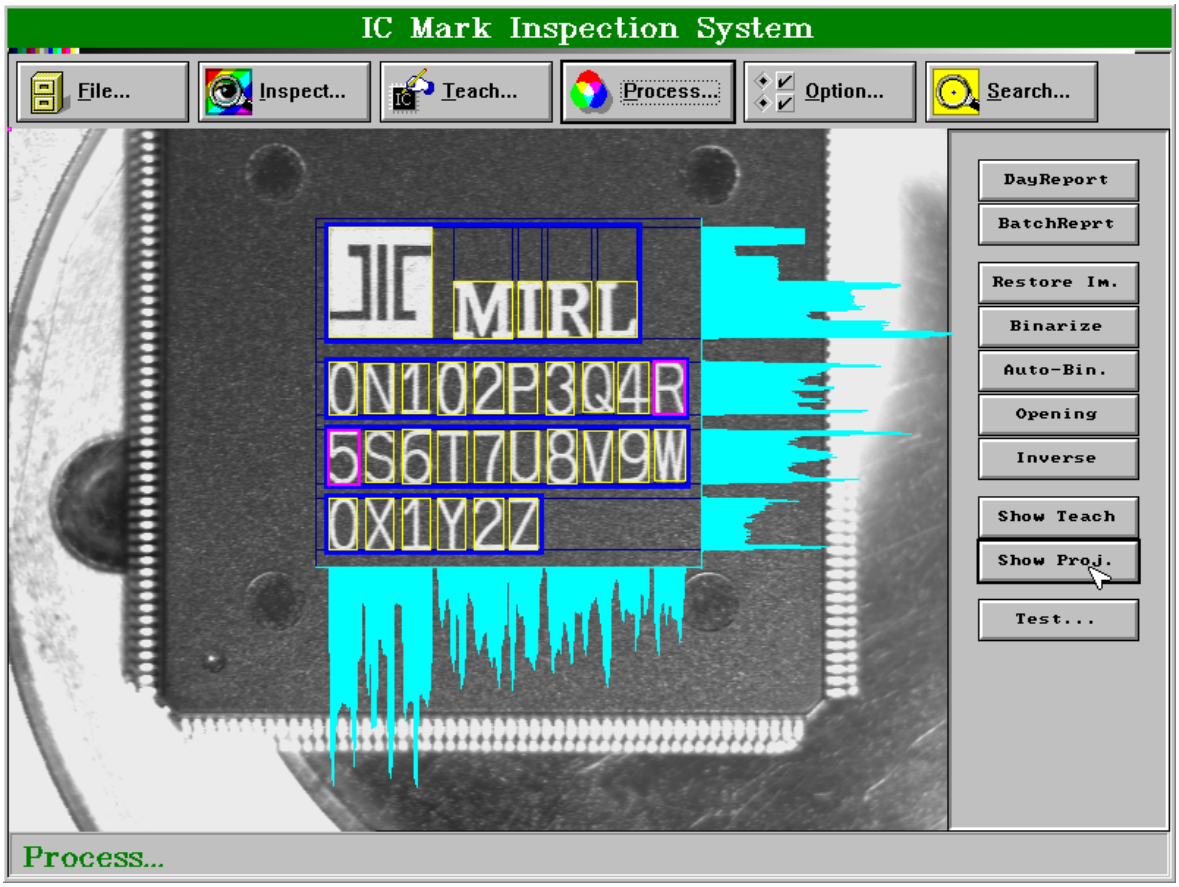

Figure 2.1 : *Use projection to segment printed characters.*

2.2 Binarizing Threshold

In order to segment the printed mark and the background of an IC chip, a proper binarizing threshold is needed. We use the minimizing within-group variance method [17] to get the best threshold of an IC image. We divide the image into two parts: the foreground and the background, and we assume each part to be Gaussian distribution. The background part has mean μ_1 and standard deviation σ_1, and the foreground part has mean μ_2 and standard deviation σ_2, as shown in Figure 2.2.

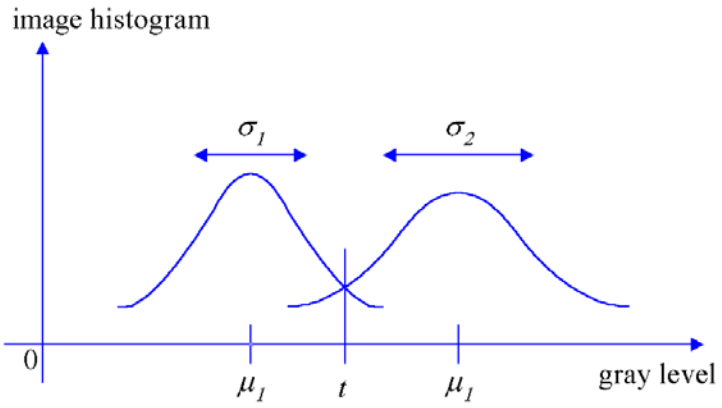

Figure 2.2: *Threshold with minimizing within-group variance method.*

Let $P(1)$,...,$P(I)$ represent the histogram of the gray value of the image. We scan threshold t at every possible gray value to get the best threshold. Let $q_1(t)$ be the probability for the group with values less than or equal to t and $q_2(t)$ be the probability for the group with values greater than t. Let $\mu_1(t)$ be the mean for the first group and $\mu_2(t)$ be the mean for the second group. Let $\sigma_1^2(t)$ be the variance for the first group and $\sigma_2^2(t)$ be the variance for the second group. The within-group variance $\sigma_w^2(t)$ is defined by

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{2.3}$$

where

$$P(1), ..., P(I): \text{ histogram probabilities}$$

$$P(i) = \frac{\#\{(r,c)|I(r,c) = i\}}{R \times C}$$

$$q_1(t) = \sum_{i=1}^{t} P(i)$$

$$q_2(t) = \sum_{i=t+1}^{I} P(i)$$

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(i)}$$

$$\mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(i)}$$

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(i)}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(i)}$$

The best threshold t can then be determined by a sequential search through all possible values of t to locate the threshold t that minimizes $\sigma_w^2(t)$. Figure 2.3 shows the thresholded result with minimizing within-group variance method.

2.3 Image Difference

Image difference is used directly to verify printed mark defect. By assuming there is no scaling between the taught data and the test IC image, translation and rotation can be solved by accurate alignment search algorithm. We perform image difference and set proper threshold to decide acceptable results and defects, then we count the defects to decide to accept or reject this IC chip. Referring to Section 2.2, we can use the minimizing within-group variance method to set the binary threshold and the difference threshold, as shown in Figure 2.2,

$$InspBinThresh = t \tag{2.4}$$

$$InspDiffThresh = \frac{|\mu_2 - \mu_1|}{2} \tag{2.5}$$

We use the average difference of the two mean values to be the difference threshold, so defect of foreground pixels will be easily separated from background pixels.

Due to the image rotation and alignment error of the inspection mechanism, some edge noise will remain after image difference, morphological opening will eliminate the noise as described in Section 2.4.

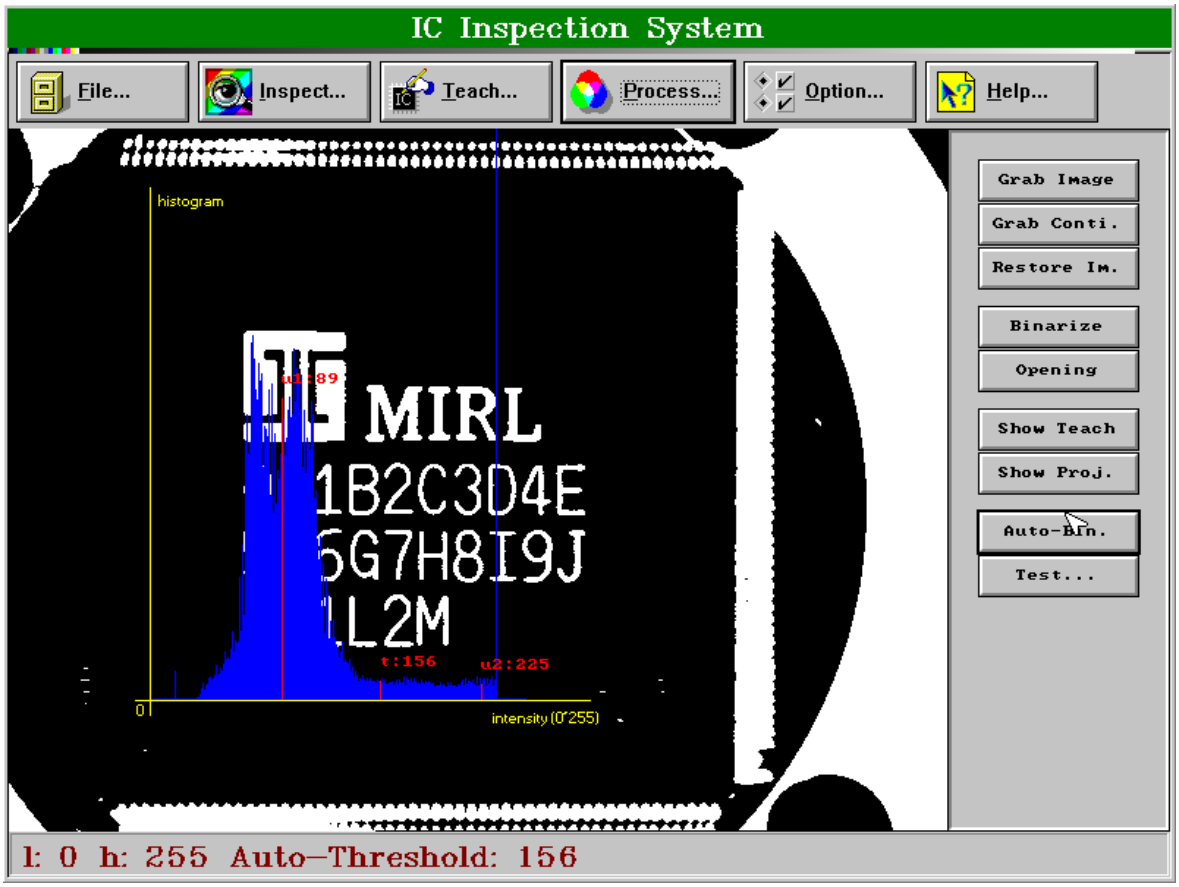

Figure 2.3: *Thresholded result with minimizing within-group variance method.*

2.4 Mathematical Morphology

Mathematical morphology [17] works on shape, and can be used to simplify image data, preserve shape, and eliminate outlier noise. Mathematical morphology includes four main operations: dilation, erosion, opening, and closing.

binary dilation of A by B: $A \oplus B$

$$A \oplus B = \{c \in E^N | c = a + b \text{ for some } a \in A \text{ and } b \in B\} \tag{2.6}$$

binary erosion of A by B: $A \ominus B$

$$A \ominus B = \{x \in E^N | x + b \in A \text{ for every } b \in B\} \tag{2.7}$$

binary opening of A by B: $A \circ B$

$$A \circ B = (A \ominus B) \oplus B \tag{2.8}$$

binary closing of A by B: $A \bullet B$

$$A \bullet B = (A \oplus B) \ominus B \tag{2.9}$$

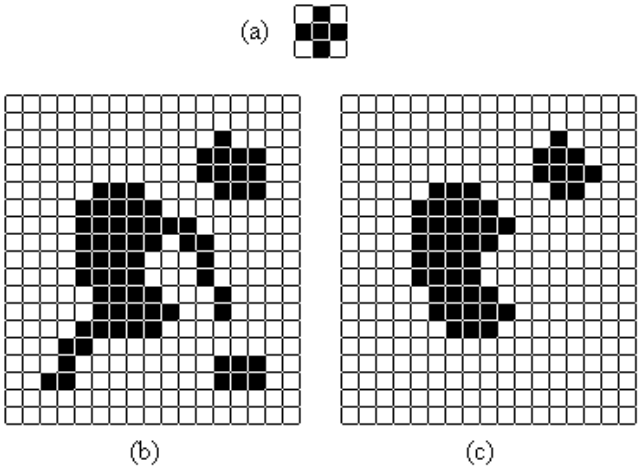

Figure 2.4 : Morphological opening and kernel. (a) The kernel. (b) Image before opening. (c) Image after opening.

The second set B is referred to as the structuring element or the kernel as shown in Figure 2.4. Grayscale morphology is similar to the binary morphology, but extends the binary value to grayscale value. First define two dual functions: top and umbra:

- top surface of A: denoted by $T[A] : F \to E$:

$$T[A](x) = \max\{y | (x, y) \in A\} \tag{2.10}$$

- umbra of f: denoted by $U[f] : U[f] \subseteq F \times E$

$$U[f] = \{(x, y) \in F \times E | y \leq f(x)\} \tag{2.11}$$

Then grayscale morphology can be defined as the following:
gray-scale dilation of A by B: $A \oplus B$

$$A \oplus B = T\{U[A] \oplus U[B]\} \tag{2.12}$$

gray-scale erosion of A by B: $A \ominus B$

$$A \ominus B = T\{U[A] \ominus U[B]\} \tag{2.13}$$

gray-scale opening of A by B: $A \circ B$

$$A \circ B = (A \ominus B) \oplus B \tag{2.14}$$

gray-scale closing of A by B: $A \bullet B$

$$A \bullet B = (A \oplus B) \ominus B \tag{2.15}$$

Morphological opening with disk kernel can eliminate sharp edge noise and preserve main shape of the image. So it will be used to do noise removal after pattern difference during inspection. It can make the inspection algorithm robust to character alignment and segmentation error of $1 \sim 2$ pixels.

Figures 2.5 and 2.6 show the inspection results before morphological opening and after opening. Outlier edge noise will be removed by the opening operation, and the defect pixels will be preserved.

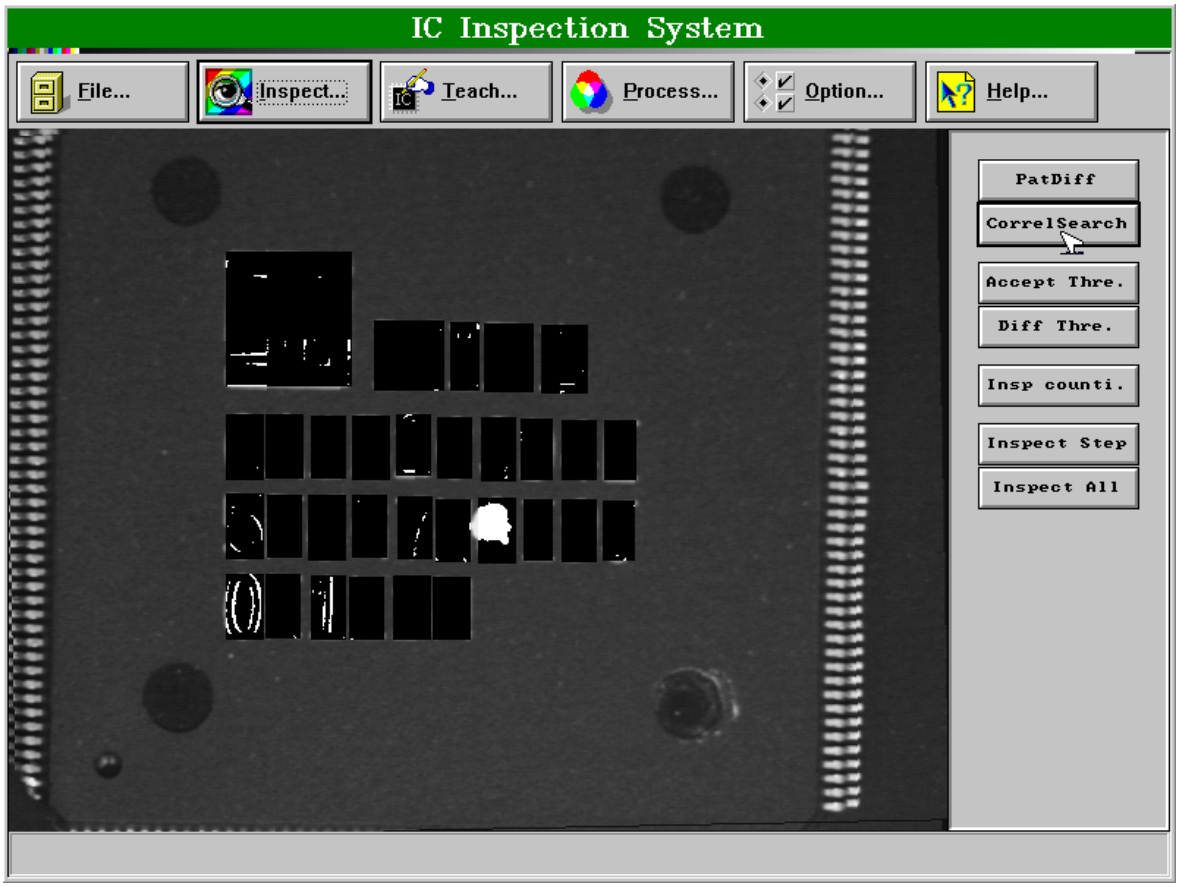

Figure 2.5: *Inspection result before opening.*

2.5 Normalized Cross Correlation

Normalized cross correlation [15] is used to match picture, i.e. match a pattern to another image and return the best matching position. The correlation coefficient $\mathbf{r}(u,v)$ is scaled in the range -1 to 1, independent of image translation and linear shifting and scaling of image gray level. The correlation coefficient is defined as follows:

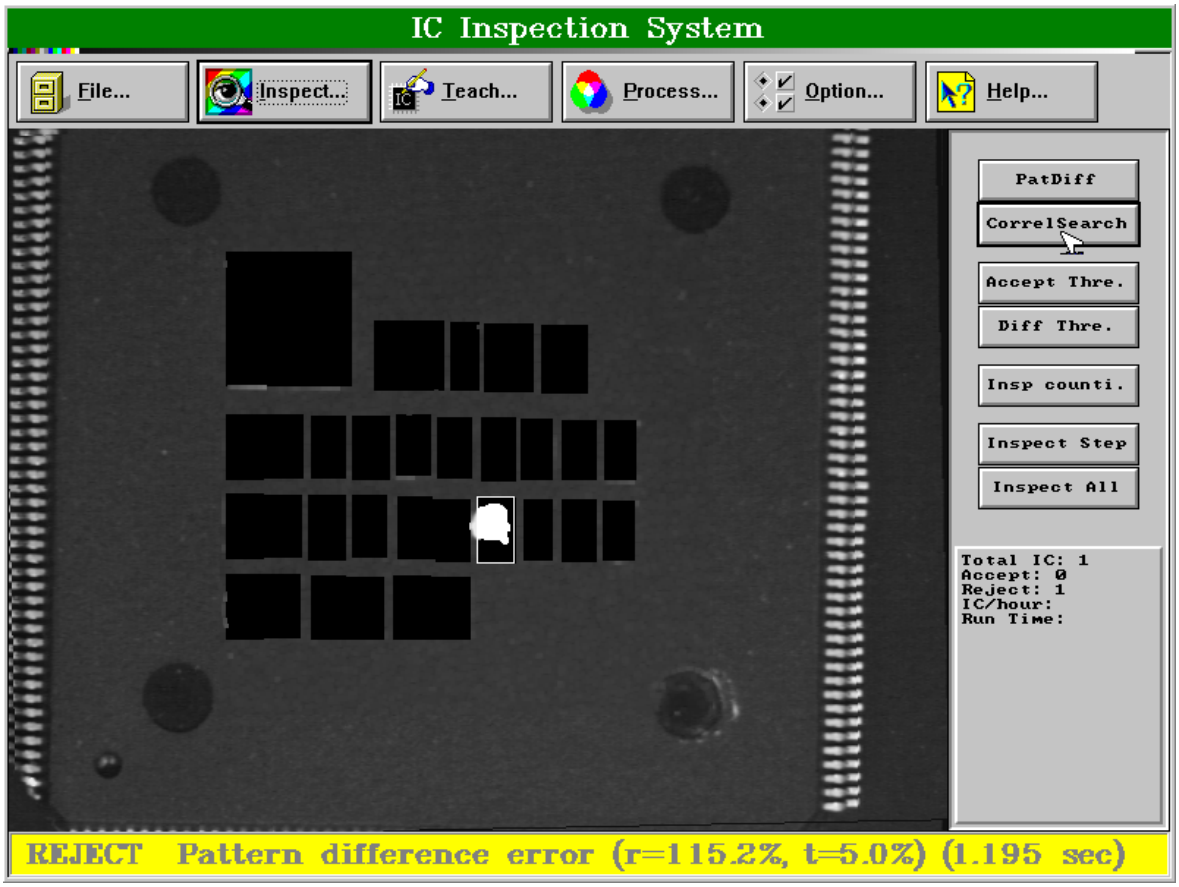

Figure 2.6: *Inspection result after opening.*

$$\mathbf{r}(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n}\left[\mathbf{I}(i+u,j+v)-\overline{\mathbf{I}}\right]\left[\mathbf{M}(i,j)-\overline{\mathbf{M}}\right]}{\sqrt{\sum_{i=0}^{m}\sum_{j=0}^{n}\left[\mathbf{I}(i+u,j+v)-\overline{\mathbf{I}}\right]^2\sum_{i=0}^{m}\sum_{j=0}^{n}\left[\mathbf{M}(i,j)-\overline{\mathbf{M}}\right]^2}} \tag{2.16}$$

where m is the pattern image width; n is the pattern image height; w is the search image width; h is the search image height; $\mathbf{r}(u,v)$ is the correlation coefficient; $\{\mathbf{M}(i,j)|0 \le i < m, 0 \le j < n\}$ is the pattern image, $\{\mathbf{I}(i,j)|0 \le i < w, 0 \le j < h\}$ is the search image, and $\{\mathbf{I}(i+u,j+v)|0 \le i < m, 0 \le j < n\}$ is the sub-image of the search image.

The output correlation coefficient is a 2D function with range

$$0 \leq u < (w - m + 1), 0 \leq v < (h - n + 1)$$

Position (u_{max}, v_{max}) is the best matching position, if $\mathbf{r}(u_{max}, v_{max}) \geq \mathbf{r}(u, v), \forall u, v$. Region $\{\mathbf{I}(i + u, j + v) | 0 \leq i \leq m, 0 \leq j \leq n\}$ is a subimage of $\mathbf{I}$ best matching image $\mathbf{M}_{m \times n}$. Coefficient $\mathbf{r} = 1$ means a perfect matching; $\mathbf{r} = -1$ means an inverse perfect matching. A perfect matching means that relation between gray level of pattern $\mathbf{M}$ and the matching sub-image $\mathbf{I}(u, v)$ is linearly shifting and scaling: $\mathbf{I}(u, v) = a\mathbf{M} + b$. An inverse perfect matching is similar to perfect matching, but has inverse gray level: $-\mathbf{I}(u, v) = a\mathbf{M} + b$.

The normalized cross correlation can be simplified as the following equations:

$$\mathbf{r}(u,v) = \frac{\sum_{i=0}^{m}\sum_{j=0}^{n}\left[\mathbf{I}(i+u,j+v)\mathbf{M}(i,j) - \mathbf{I}(i+u,j+v)\overline{\mathbf{M}} - \overline{\mathbf{I}}\mathbf{M}(i,j) + \overline{\mathbf{I}}\overline{\mathbf{M}}\right]}{\sqrt{\sum_{i=0}^{m}\sum_{j=0}^{n}\left[\mathbf{I}^2(i+u,j+v) - \mathbf{I}(i+u,j+v)\overline{\mathbf{I}}\right]}\sqrt{\sum_{i=0}^{m}\sum_{j=0}^{n}\left[\mathbf{M}^2(i,j) - \mathbf{M}(i,j)\overline{\mathbf{M}}\right]}} \tag{2.17}$$

We multiply both the numerator and the denominator by factor $\mathbf{N} = m \times n$:

$$\mathbf{r}(u,v) = \frac{\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\mathbf{M}(i,j) - \mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\overline{\mathbf{M}}}{\sqrt{\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\left[\mathbf{I}^2(i+u,j+v) - \mathbf{I}(i+u,j+v)\overline{\mathbf{I}}\right]}\sqrt{\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\left[\mathbf{M}^2(i,j) - \mathbf{M}(i,j)\overline{\mathbf{M}}\right]}} \tag{2.18}$$

$$\mathbf{N}\overline{\mathbf{M}} = \sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}(i,j) \tag{2.19}$$

$$\mathbf{N}\overline{\mathbf{I}} = \sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v) \tag{2.20}$$

Constants $\overline{\mathbf{I}}$ and $\overline{\mathbf{M}}$ can be moved out: $\mathbf{r}(u,v) =$

$$\frac{\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\mathbf{M}(i,j) - \left(\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\right)\left(\mathbf{N}\overline{\mathbf{M}}\right)}{\sqrt{\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}^2(i+u,j+v) - \left(\mathbf{N}\overline{\mathbf{I}}\right)\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)}\sqrt{\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}^2(i,j) - \left(\mathbf{N}\overline{\mathbf{M}}\right)\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}(i,j)}} \tag{2.21}$$

We get the final simplified equation of the normalized cross correlation: $\mathbf{r}(u,v) =$

$$\frac{\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\mathbf{M}(i,j) - \left(\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\right)\left(\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}(i,j)\right)}{\sqrt{\left[\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}^2(i+u,j+v) - \left(\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\right)^2\right]\left[\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}^2(i,j) - \left(\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}(i,j)\right)^2\right]}} \tag{2.22}$$

where $\mathbf{N} = m \times n$.

The simplified equation can be accelerated by a dynamic programming method, which will be introduced at Chapter 3. The normalized cross correlation is one of the best method to match images, but the computational complexity of the traditional correlation is very time consuming. Our proposed method can solve the problem and will be introduced in Chapter 3.

Chapter 3

Fast Search Algorithms

3.1 Introduction

For visual inspection, high precision alignment enables efficient implementation of defect inspection; it is also the key to high speed and high accuracy verification. But fast alignment algorithm is almost the most difficult part of visual inspection, because the accuracy, functionality, reliability, repeatability, and speed can not be satisfied at the same time. For different purposes of application, the requirement of alignment is distinct. Ad hoc methods may be used to speed up the computation, but the general purpose fast search algorithm is still an important problem in computer vision and artificial intelligence.

Our purpose is to design a general purpose fast search algorithm which can match a 128×128 grayscale pattern in a 640×480 field of view. The computing time is within 50 to 70 milli-second (ms), and the locating accuracy can achieve subpixel accuracy.

Normalized cross correlation is our main idea to perform general purpose matching. We improve the speed, preserve the excellent property, and enhance the ability of normalized correlation search. In addition to the speed and accuracy requirement, our search algorithm can perform multiple target search i.e. return matching points sorted

in the descending order of similarity. In addition, the automatic model selection is also an important role in our search function.

The normalized correlation search has many excellent properties for alignment [38]:

- Independence of linear image brightness changes
  - independence of brightness shift
  - independence of brightness scaling
- Absolute value of matching (correlation coefficient)
- Typical sub-pixel accuracy: 0.1 to 0.25 pixel
- Excellent repeatability: $\sigma_{position} \leq 0.05$ pixel
- Excellent immunity to image defects
- Tolerance of image distortion due to rotation or scaling
- Excellent discrimination ability
- Tunability to appropriate feature scale
- Excellent immunity to scene clutter
- Computational efficiency!

We improved the original normalized cross correlation by using dynamic programming method and resolution pyramid search. The resolution pyramid search i.e. hierarchical search is an efficient method to divide brute force search into a global search and a local search. By hierarchical sub-sampling of the pattern and search image, we get two sets of resolution pyramid images. By full search on the smallest sub-sampling image, we reduce the global search space. After getting the global matching point, we

perform one or more local search on the finer resolution image layer near the neighborhood. Repeating this method, we can get the accurate matching point of the original image. We can enhance this method to over-samplimg the image to get the subpixel matching accuracy. The fast search algorithm will be described in Sections 3.2, 3.3, and 3.4.

Normalized correlation search returns an absolute value of similarity of each matching point, so we can sort the matching scores in descending order. By selecting maximum matching points or minimum matching score of each matching layer, we can get multiple matching targets. Section 3.5 will explain the method in detail.

The automatic detection of search model is an important part of an intelligent alignment function. By specifying width and height of the model and the search range, the program can return the best model for alignment. The best model will be unique, discernible, discriminable, and without ambiguity. We use the property of similarity of correlation score and the image property to decide the best model, as shown in Section 3.6.

Section 3.7 gives some conclusions on the search algorithms. Comparison with the other search methods and the coding techniques will be also mentioned here. Section 3.8 will show some experimental results.

3.2 Dynamic Programming to Speed up Normalized Cross Correlation

Referring to Equation 2.22, seven main terms constitute the correlation equation. As shown in the Equation 3.1, the three terms about the pattern image **M** can be computed in the pre-processing step. The three terms about the search image **I** can be computed once by the dynamic programming (D.P.) method in the pre-processing step. The dominating term of the correlation of the pattern and the search image is of the

complexity $O(m \times n \times w \times h)$, i.e. computing the correlation of one $m \times n$ image and another $w \times h$ image needs time about $m \times n \times w \times h$.

$$\mathbf{r}(u,v) = \frac{\overbrace{\mathbf{N}\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\mathbf{M}(i,j)}^{\text{Dominator}} - \overbrace{\left(\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)\right)}^{\text{D.P.}\sum\text{im}}\overbrace{\left(\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}(i,j)\right)}^{\sum\text{pim}}}{\sqrt{\left[\mathbf{N}\underbrace{\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}^2(i+u,j+v)}_{\text{D.P.}\sum\text{im}^2} - \left(\underbrace{\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{I}(i+u,j+v)}_{\text{D.P.}\sum\text{im}}\right)^2\right]\left[\mathbf{N}\underbrace{\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}^2(i,j)}_{\text{D.P.}\sum\text{pim}^2} - \left(\underbrace{\sum_{i=0}^{m}\sum_{j=0}^{n}\mathbf{M}(i,j)}_{\sum\text{pim}}\right)^2\right]}} \tag{3.1}$$

where $\mathbf{N} = m \times n$.

To compute the dominating term of the normalized correlation, we have three main approaches. First, we use the resolution pyramid to reduce the correlation space, and it will be described in Section 3.3. The second is about the coding techniques and the hardware approaches. We can use inline assembly to optimize the kernel code of the computation, we can also take advantage of the Intel MMX Technology to accelerate the computation. Third, the Fast Fourier Transform (FFT) is a good approach to reduce the computation of $O(n^4)$ to $O(n^2 \log n^2)$, but the gain of time is not worthwhile. Because after resolution sub-sampling, image size is quite small so that the FFT is not a good approach here. Other acceleration methods will be discussed in Section 3.7.1.

The original normalized correlation $\mathbf{r}(u,v)$ needs a square root computation and ranges between -1 and 1, we discard the inverse matching and define the new correlation factor $\mathbf{CF}(u,v)$:

$$\mathbf{CF}(u,v) = \max(\mathbf{r}(u,v),0)^2 \tag{3.2}$$

$$0 \le \mathbf{CF}(u,v) \le 1$$

By computing $\mathbf{CF}(u, v)$ instead of $\mathbf{r}(u, v)$, we avoid the square root of a floating number and discard the inverse matching, and the range is between 0 and 1.

The dynamic programming approach to reduce the computation of the correlation will be explained here. As shown in Figure 3.1 and Equation 3.3, we sum up pixels and construct the table step by step:

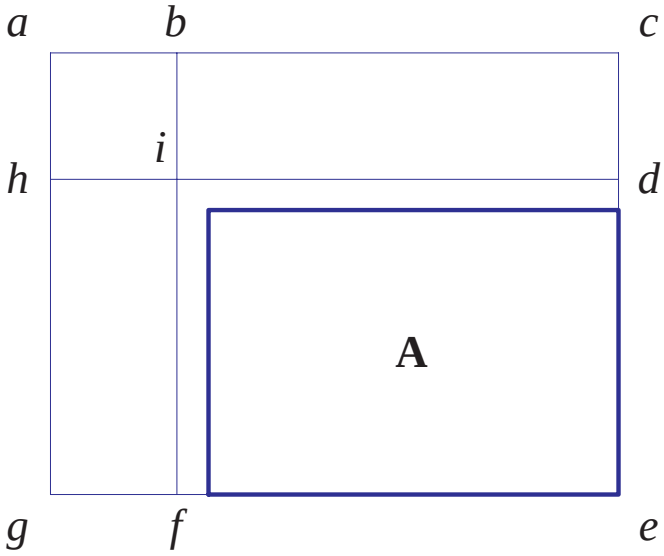

Figure 3.1 : *Use dynamic programming table to compute the sum of pixels of a specific area.*

$$\mathbf{A} = aceg - abfg - acdh + abih = e - f - d + i \tag{3.3}$$

where *aceg*, *abfg*, *acdh*, and *abih* are sums of pixels of the rectangle area; *e*, *f*, *d*, and *i* are sums of all upper-left pixels of the points.

The initialization and construction of the dynamic programming table is in the similar way, as shown in Figure 3.2 and Equation 3.4:

$$R = P + Q - S + r \tag{3.4}$$

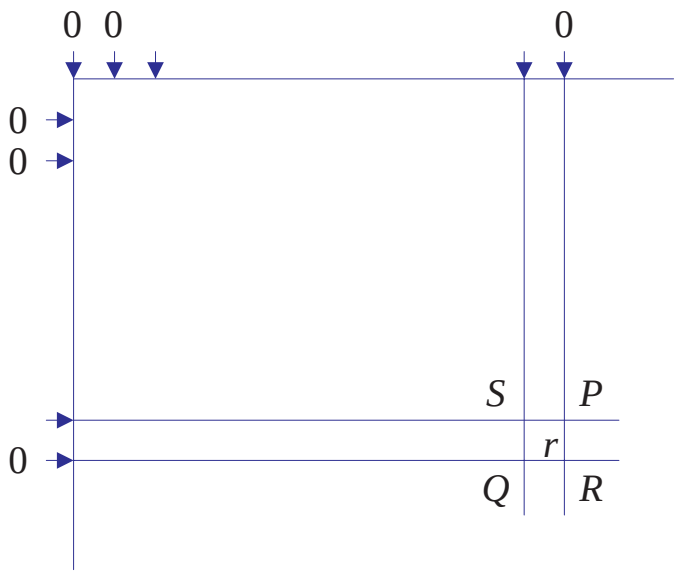

Figure 3.2: *The construction of the dynamic programming table.*

where r is the pixel value of the location R; S, P, Q, R are values from D.P. table at that position.

With the dynamic programming table, computing pixel sum of any specific area requires only two subtractions and one addition. The table is easy to compute at the pre-processing step before search.

Due to the nature of the pixel summing table, overflow is an important problem to avoid. Multiple tables with unsigned long integer will solve this problem. Section 3.8 shows some experimental results of the search procedure. Figures 3.11, 3.12, 3.13, 3.14, and 3.15 show the search images and the corresponding correlation function maps.

3.3 Resolution Pyramid Search/Hierarchical Search

Resolution pyramid or hierarchical search is an important solution to the heavy computation of normalized correlation search. It not only can preserve almost all excellent properties of normalized correlation search, such as tolerance to linear change of image

intensity, accuracy, reliability, and returning an absolute similarity score, as stated in Section 3.1; it is also fast and easily extensible to subpixel accuracy. To achieve fast search, we need image sub-sampling; on the other hand, in order to search accurately, we need image over-sampling. We sort these pattern and search image layers to a pyramid structure. The number of layers is adjustable, and depends on the pattern and the search image sizes. The global search is only performed on the coarse layer, and the local search and subpixel search are on the finer layers.

3.3.1 Image Sub-Sampling

We use the average of four pixel values as the sub-sampling pixel, as shown in Figure 3.3. With proper coding technique, we can use two pointers p_1 and p_2 and shift instruction in assembly language to achieve the work quickly. Figure 3.4 shows the average method and the upper-left pixel method. In multiple layers of pyramid, we choose the average method for stability.

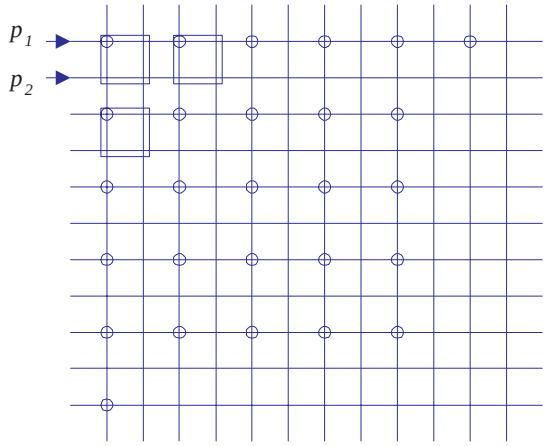

Figure 3.3 : *Image sub-sampling, the average method and the upper-left pixel method.*

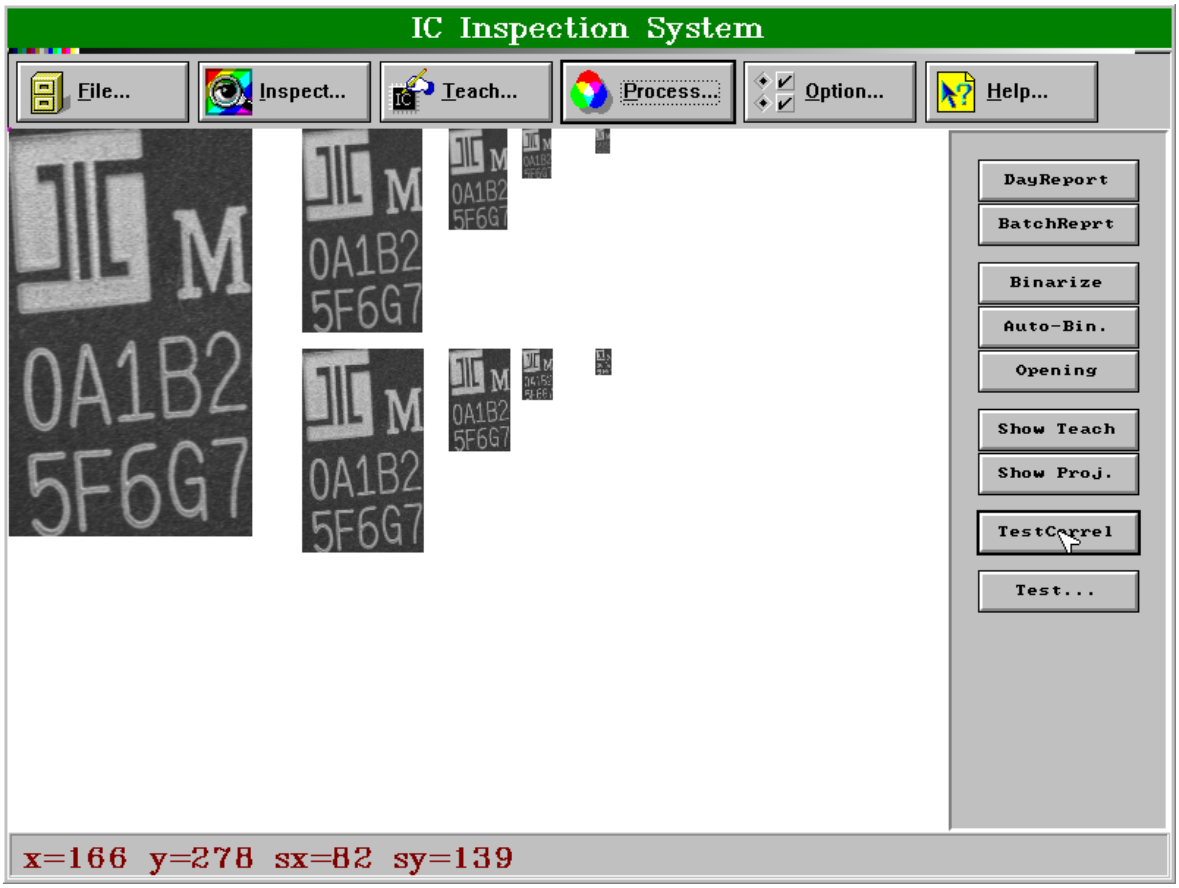

Figure 3.4 : *Comparison of the two sub-sampling methods: The upper images show the average method, and the lower images show the upper-left pixel method. The outcome result of sub-sampling shows that the average method gets the better effect.*

3.3.2 Hierarchical Search

We match the pattern and the search image layers in pair, and define the top layer as the smallest sub-sampled layer. We perform coarse search in the top layer, and perform fine search in the lower layers. The coarse search is a full global search to detect the coarse position. The fine search is a local 3×3 search near the coarse position in the next lower layer, as shown in Figure 3.5. We continue searching in the finer layers until the last layer is reached, or continue to search in the over-sampled subpixel layer.

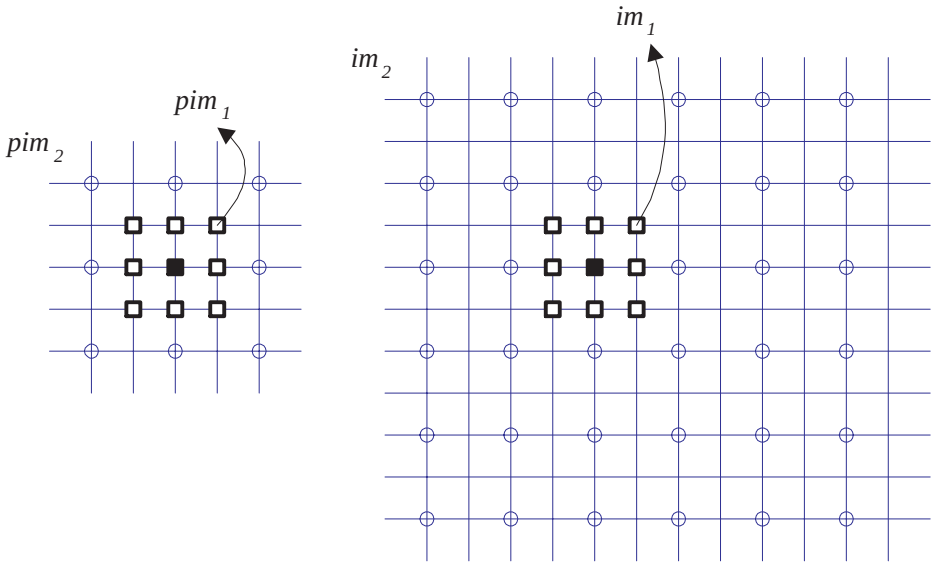

Figure 3.5 : *The fine search in the resolution pyramid layer. The coarse search matches pattern pim_2 in image im_2, and the fine search matches pattern pim_1 in the neighborhood of image im_1. Images im_2 and pim_2 are sub-sampling of im_1 and pim_1 respectively.*

Each sub-sampling layer speeds-up the coarse search by the factor of $2^4 = 16$. Let the total pyramid layer be r, the coarse search is faster than the original correlation search by the speed-up factor of $16^{(r-1)}$. The computing time of the fine search depends on r and the pattern image size.

	the coarse search	the fine search
search for a large pattern in a large range	high speed-up factor (high pyramid layer)	time consuming (dominating)
search for a small pattern in a large range	low speed-up factor (low pyramid layer)	fast

Table 3.1 : *Speed comparison of the hierarchical pyramid search.*

Consider the search time in the following two conditions, as shown in Table 3.1:

- search for a large pattern in a large range

  The pattern and the search range are large enough to make many pyramid layers, so the coarse search is fast; but it takes much time to do the fine search through the finer layers.

- search for a small pattern in a large range

  The pattern is small that we can not get too many pyramid layers, so the coarse search can not get benefits from it; but the fine search will be fast due to the few layers and small pattern.

We can conclude that the computing time of resolution pyramid search depends on the resolution layer, and the resolution layer depends on the accuracy and information that sub-sampling image preserves. Section 3.3.3 will discuss our adaptable resolution pyramid search method based on the conclusion.

```
min = minimum (pwidth, pheight);
if (min<12) RLayer=1;
else if (min<24) RLayer=2;
else if (min<48) RLayer=3;
else if (min<96) RLayer=4;
else if (min<192) RLayer=5;
else if (min<384) RLayer=6;
else RLayer=7;
```

Table 3.2 : *The pseudo code that decides the number of resolution layer (RLayer) from pattern width (pwidth) and pattern height (pheight).*

3.3.3 Adaptable Resolution Pyramid Search

In order to get the best search speed, we must get the most resolution layers and the sub-sampling pattern that do not lose too much accuracy or information. The number of layers depends on the pattern and search image sizes, and it will affect the accuracy and the computation time.

We decide the number of layer based on the width and height of the pattern image. We choose the smaller value of the pattern width and height to be the criterion. Table 3.2 shows the pseudo code that decides the number of resolution layer. The boundary value of the table is calculated from Table 3.3. We force the image width to be even for ease of computation. Figure 3.6 shows the example of resolution pyramid search.

3.4 Subpixel Search

It is easy to extend the resolution pyramid method to achieve the subpixel accuracy by over-sampling the pattern and search image. Typically 0.5 and 0.25 subpixel accuracy

min	*RLayer*	*pwh*	*min*	*RLayer*	*pwh*
1-8	1	2,4,6,8	80-95	4	10
9-11	1	10	96-127	5	6
12-15	2	6	128-159	5	8
16-19	2	8	160-191	5	10
20-23	2	10	192-255	6	6
24-31	3	6	256-319	6	8
32-39	3	8	320-383	6	10
40-47	3	10	384-511	7	6
48-63	4	6	512-639	7	8
64-79	4	8	640-767	7	10

Table 3.3 : *The calculation of the layer boundary where pwh means the pattern width or height in the top layer.*

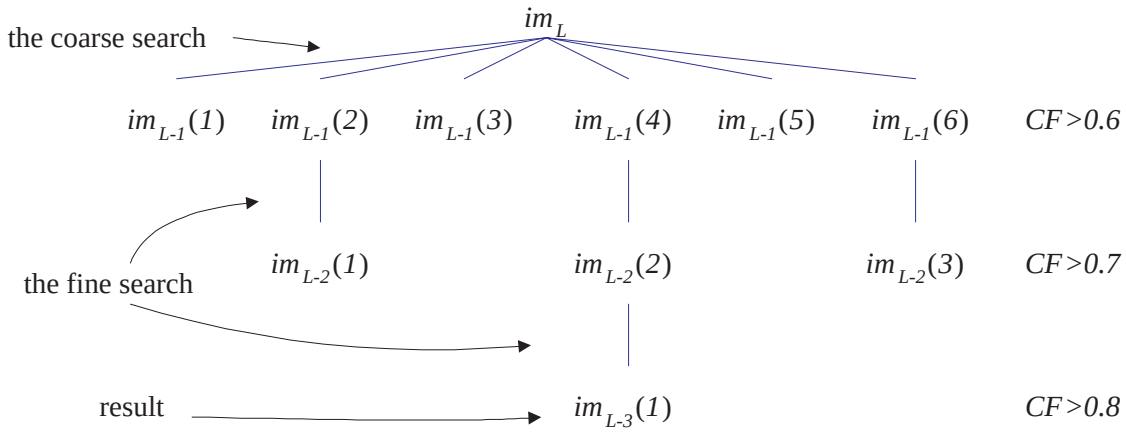

Figure 3.6 : *Example of the resolution pyramid search. After coarse search, we get six candidates with correlation factor* $CF > 0.6$ *to perform the fine search, then we get three candidates with* $CF > 0.7$ *to perform the finer search. At last we get the matching point with* $CF > 0.8$ *as the result.*

is easy to obtain and it does not affect the computation time too much. To compute the subpixel matching in the over-sampling layer will consume time by a factor of four of the previous layer.

3.4.1 Image Over-Sampling

We over-sample the image by bi-linear interpolation, as shown in Figure 3.7. The example of over-sampled images is shown in Figure 3.8.

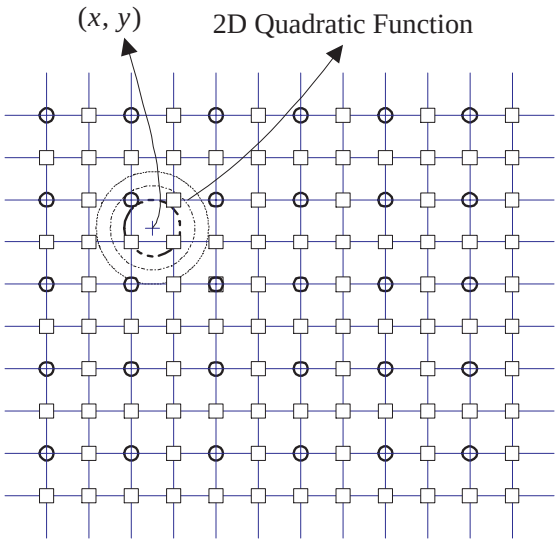

Figure 3.7 : *Image over-sampling and subpixel interpolation. Image over-sampling can achieve 0.5 or 0.25 subpixel accuracy typically. We use the 2D quadratic function to model the correlation function near the matching point, then we can calculate the more accurate matching point* (x, y).

3.4.2 Estimation of the Accurate Matching Point

Referring to Figure 3.7, we can use the 2D quadratic interpolation to estimate the more accurate subpixel matching point. Let the desired best matching point be (x, y), and

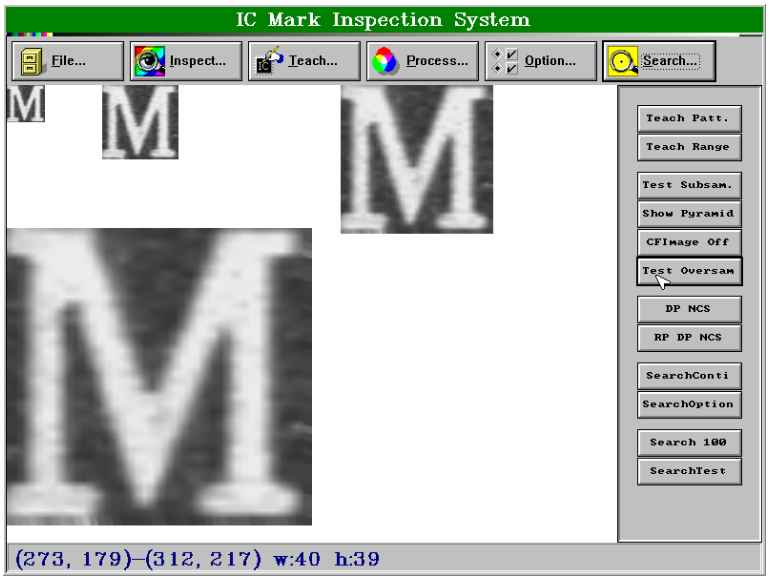

Figure 3.8 : *Example of image over-sampling. The source image is shown in the upper-left corner. The first order, second order, and third order over-sampled images are shown to the right and below.*

let the 2D correlation function near the point be quadratic:

$$f(x, y) = a + bx + cy + dx^2 + exy + gy^2 \tag{3.5}$$

To solve the six unknowns a, b, c, d, e, and g, we need six points near the matching point to solve Equation 3.5. As we get the function parameters a, b, c, d, e, and g, we can calculate the accurate matching point. The two directional differentials of slope near the local maximum will be zero, so we get the two simultaneous equations:

$$\frac{\partial f}{\partial x} = b + 2dx + ey = 0 \tag{3.6}$$

$$\frac{\partial f}{\partial y} = c + ex + 2gy = 0 \tag{3.7}$$

We can solve the two unknowns x and y by the two simultaneous equations and get the accurate subpixel matching. We will add this function to our program and improve

the subpixel matching accuracy in the future.

The experimental results of the subpixel search will be shown in Section 3.8.

3.5 Multiple Target Search

In many applications, the matching object in the search range will not be unique, so multiple target search is a practical and important function. After calculating the correlation function, we can use selection sort to pick up the most matching targets sorted in the descending order. In our multiple level pyramid search, proper threshold value for both the coarse search and the fine search is needed. We will explain the different options of our search function in Table 3.4.

3.6 Automatic Search Model Detection

The automatic detection of search model is an important function in automatic alignment solution. Model selection by human is subjective and may lead to error. In industrial application, the operator may not know about the design of the fast search algorithm in detail, so the automatic model selection is especially meaningful and indispensable.

We can design the model selection algorithm according to the search method. The coarse search is global and determines the search result, so we will select the model using the top sub-sampled layer. Thus we can get the better model detection speed, and the model is robust for the sub-sampling search.

We define the input and output of the model detection algorithm as follows. The inputs are the pattern width, pattern height, and the search range. The number of candidates to be found is also a specified parameter. The outputs are the positions of the best model candidates sorted in the descending order.

COARSE_SEARCH		coarse search only
SEARCH_ONE_FAST		search for one target with early jump out
SEARCH_ONE	*nCoarseTarget*=3 *nSearchTarget*=1	search for one target exactly and robustly
SEARCH_MN	*nCoarseTarget*=10 *nSearchTarget*=5	coarse search for 10 targets and fine search for 5 targets among them
SEARCH_MC	*dCoarseMinCF*=0.2 *dSearchMinCF*=0.5	coarse search for $CF \geq 0.2$ and fine search for $CF \geq 0.5$ among them
AUTO_SEARCH	*dCoarseMinCF*=0.2 *nCoarseTarget*=10 *dSearchMinCF*=0.5	automatic search, the default parameters are used

Table 3.4 : *Multiple target search methods and parameters. Option COARSE_SEARCH is to perform coarse search only. Option SEARCH_ONE_FAST is the most popular search option to match a unique target quickly. Option SEARCH_ONE will search for three targets in the coarse layer and find the final matching point among the three candidates in the fine layers. Option SEARCH_MN searches for multiple targets by specifying the number of coarse search targets and fine search targets. Option SEARCH_MC searches for multiple targets by specifying the minimum matching scores of both the coarse and fine searches. Option AUTO_SEARCH searches for multiple targets using the default parameters. We can specify different search options for different applications.*

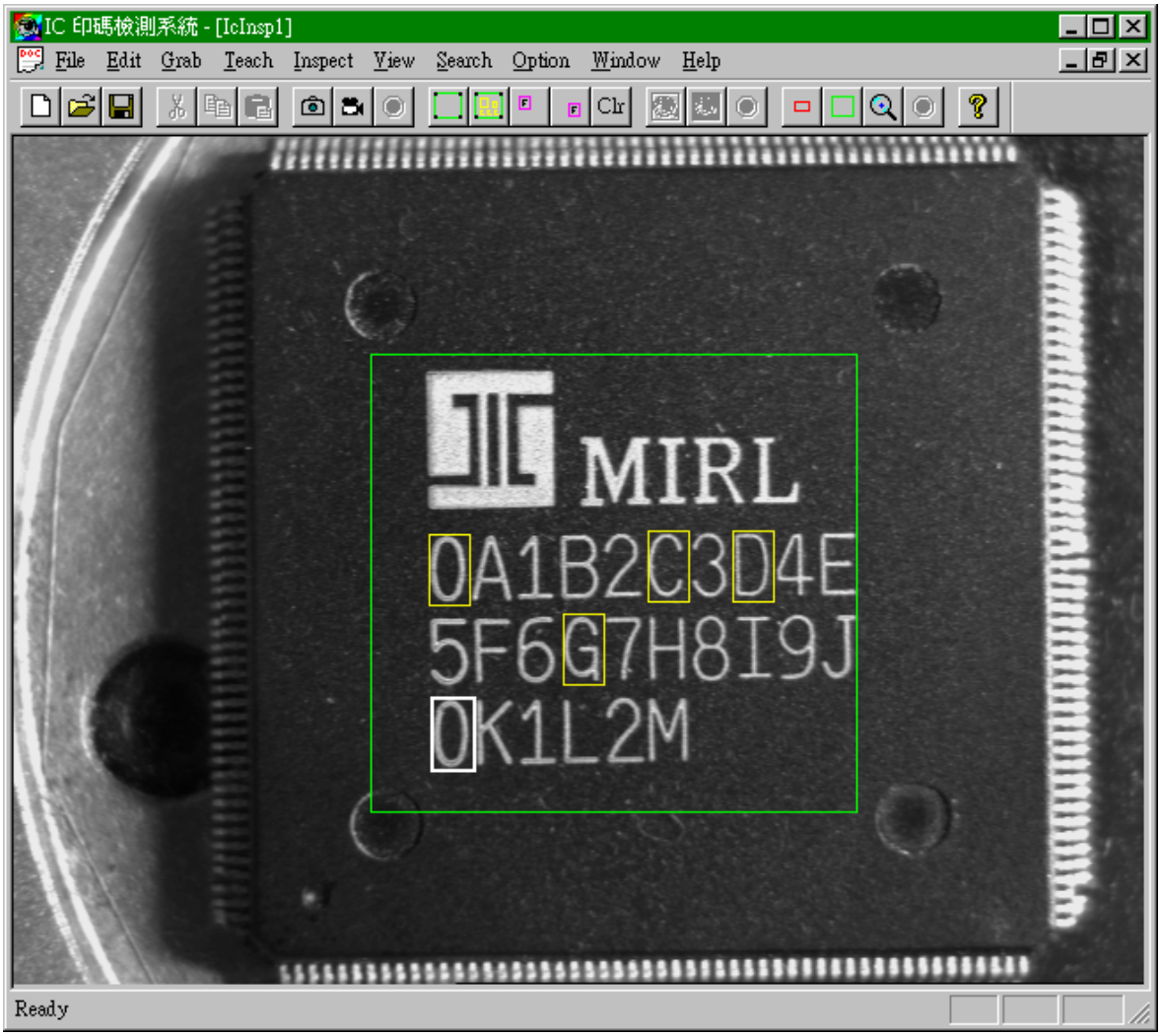

Figure 3.9 : *Example of the multiple target search on an IC image. The pattern is the printed mark '0' at the lower left corner. As a result, five targets are returned. The printed mark '0' at the upper left corner gets the higher matching score, and the printed marks 'C', 'D', and 'G' get the lower matching score. (SEARCH_MN, dCoarseTarget=10, dSearchTarget=5)*

We design the model selection algorithm by calculating the *position score* for local uniqueness and the *uniqueness score* for global uniqueness [10]. The *position score* is defined by the *variance score* multiplied by the local position uniqueness, and the *uniqueness score* is defined by the global uniqueness, as shown in Equations 3.8 and 3.9.

$$position\ \ score = variance\ \ score \times (1 - max8CF) \tag{3.8}$$

$$uniqueness\ \ score = 1 - secondCF \tag{3.9}$$

where the $max8CF$ is the largest correlation score among the eight neighbors without the position itself, and the $secondCF$ is the largest correlation score in the search range without the best match itself. Because the correlation score is between 0 and 1, $1 - max8CF$ represents the local position uniqueness and $1 - secondCF$ represents the global uniqueness. The scene contrast is also important information for model candidate selection, so the *variance score* will be taken into consideration. For the ease of computing, we define the *variance score* by the difference of the highest gray level and the lowest gray level of the model candidate.

The automatic model selection algorithm is described as follows:

1. For each position of the search range, calculate the position score.
2. Use selection sort to get the maximum n_c candidates in the descending order.
3. For the n_c candidates, search globally to calculate the uniqueness score.
4. Use selection sort to get the maximum n_m models in the descending order.
5. Return the positions of the n_m models.

Figure 3.10 shows the example of automatic search model detection (n_c=25, n_m=5).

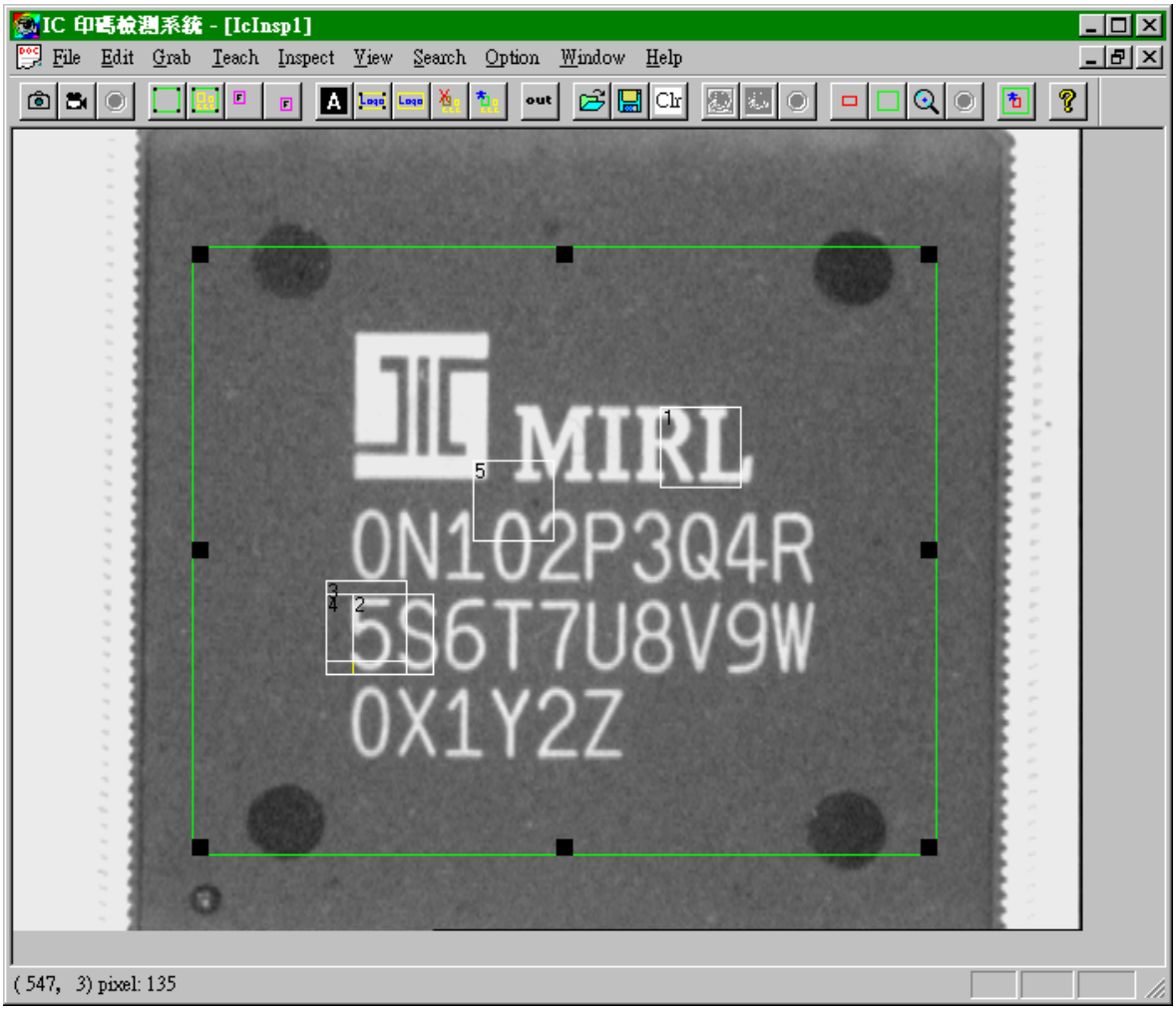

Figure 3.10 : *Example of the automatic detection of search model. The returned model positions which are unique in the search range are shown in the numbers from 1 to 5.*

3.7 Conclusion on Fast Search Algorithms

Conclusion and discussion of the search methods will be presented here. Implementation details of the search method will also be stated in this section.

3.7.1 Comparison with Other Search Methods

We collect various methods to develop our search algorithm. Since matching is an important algorithm in theoretical and industrial areas, there are many other approaches to implement matching, but it is impossible to gather all these methods into a single algorithm. Trade-off among accuracy, functionality, and speed is always an important consideration. Ad hoc methods for special applications can not satisfy the general purpose search requirement. We will describe the other methods and compare them with the proposed method here.

- Sum of Absolute Difference

  Sum of absolute difference (SAD) is a simple and fast method for matching. SAD has the excellent hardware acceleration benefit for implementation, by the help of Intel MMX technology, we can process 64-bits in a single instruction. Furthermore, early jump-out [11, 42] is easily added to the SAD method and dramatically increases the matching speed. But the SAD can not return an absolute value of matching and can not handle linear transformation of image intensity, so the match functionality is much poorer than the NCS, as discussed in Section 3.1. The tolerance for image rotation and noise, the subpixel accuracy, and the repeatability of SAD are also much poorer than the NCS, so we discard the SAD and choose the NCS as our main matching method.

- Fast Fourier Transform

  Fast Fourier Transform (FFT) is an important method to decrease the order of

convolution from $O(n^2)$ to $O(n \log n)$, where n is the pixel number, as discussed in Section 3.2. But the transformation requires numerous floating point calculations, and it is not worthwhile for small sub-sampled image, so the FFT is not feasible to improve our method.

- Matching by Edge or Skeleton

  Matching by the edge or skeleton is also a feasible method to reduce the full search. But the performance is highly dependent on the feature extraction method, and it may lead to unstable matching results. The edge extraction and the smoothness of the image will also affect the performance, and the edge or skeleton extraction is hard to speed up, so we discard the method.

- Horizontal or Vertical Projection to Match First

  This is an ad hoc search method for application with no rotation. Advantage of the method is to reduce the 2D matching into two 1D matching, but the disadvantage is the degradation. It is not robust to slight rotation, distortion, and noise.

- Binarized or Color Down-Sampled Image to Match First

  Using binarized image to search first is a good way similar to SAD, but the gray level information is discarded, and the benefits of NCS is lost. The color down-sampling is suitable for color image with multiple bands and is not applicable for 8-bit grayscale image.

We can conclude that our search method derives the most benefits of the matching functionality, accuracy, speed, and robustness among these methods. With proper coding techniques, we can get the most efficiency of our algorithm. Section 3.7.2 will describe these techniques.

3.7.2 The Coding Techniques and the Implementation

Coding techniques of our fast search algorithm will be mentioned here. Since the design of the algorithm and the choice of the best resolution layer is critical to the search performance, the implementation technique is still important, too. With the help of the dynamic programming method, speedup of the correlation kernel is the primary target to optimize the search. By accommodation to the machine architecture, we achieve the most efficiency by considering both the functionality and the implementation feasibility of the following approaches:

- Kernel Code Optimization

  By optimizing the kernel code of the search procedure, we get large improvements on speed. We write the code in the style that compiler can translate the optimized code [4]. Then we watch the assembly code generated by the compiler and perform partial optimization on critical parts using inline assembly if necessary.

- Using 1D Sequential Array Instead of 2D Array

  We store the image in the 1D sequential buffer, so we can take the advantages from the fast sequential memory access and the cache benefits.

- Checking Data Flow among the CPU, Data Bus, Frame Grabber, and the CCD Camera

  We should always check that the data flow among these components is optimized. So the throughput of the system will be maximized.

- The Intel MMX Technology

  The Intel MMX technology is the new extension to multimedia and signal processing calculation with Single-Instruction-Multiple-Data (SIMD) ability. The MMX SIMD instruction can calculate sum of multiplication of four integers to four integers at a single instruction and dramatically increase the correlation

computation. We will add the MMX coding approach to our search procedure in the future.

3.7.3 Future Direction

Future direction of our research on fast search algorithms will focus on the ability to handle image rotation, scaling, and slightly distortion and transformation. The subpixel accuracy of our method can also be improved.

By way of novel feature extractions and artificial intelligence methods instead of full search, we can improve the search algorithm and achieve the more powerful and intelligent matching in computer vision application.

3.8 Experimental Results

Figures 3.11, 3.12, 3.13, 3.14, and 3.15 show the experimental results of the search procedure and the corresponding correlation function maps. Figures 3.16 and 3.17 show the two targets successfully found in an under-lighted image and an over-lighted image with options *SEARCH_MN*, *nCoarseTargets*=3, $dCoarseMinCF = 0.2$, $nSearchTargets = 3$, and $dSearchMinCF = 0.5$. Figure 3.18 shows two correct targets and one similar target are found in a rotated and under-lighted image with the same options. The search condition is not stable here, but candidate positions similar to the pattern still get the matching score greater than 0.5. The best matching position gets the score of 0.518.

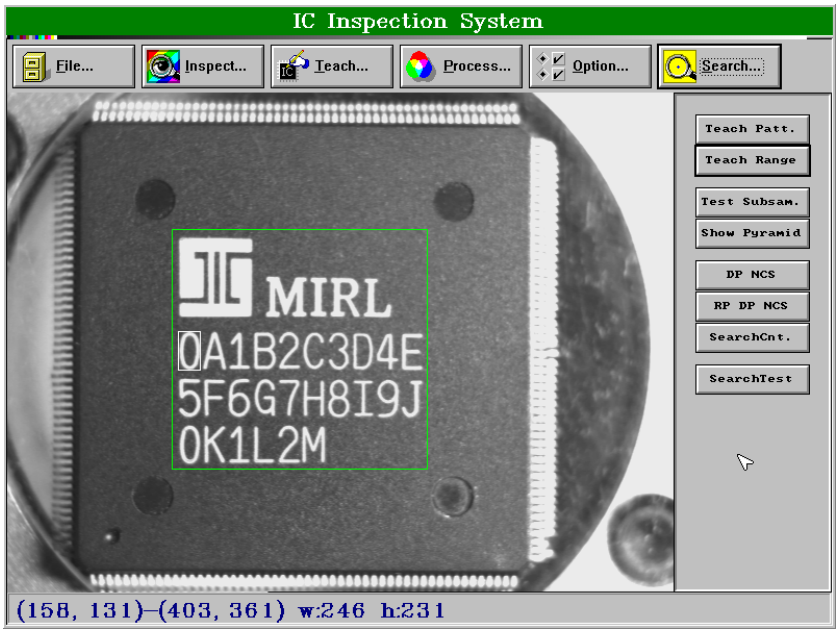

Figure 3.11 : *The pattern is the small rectangle, and the search range is the big rectangle.*

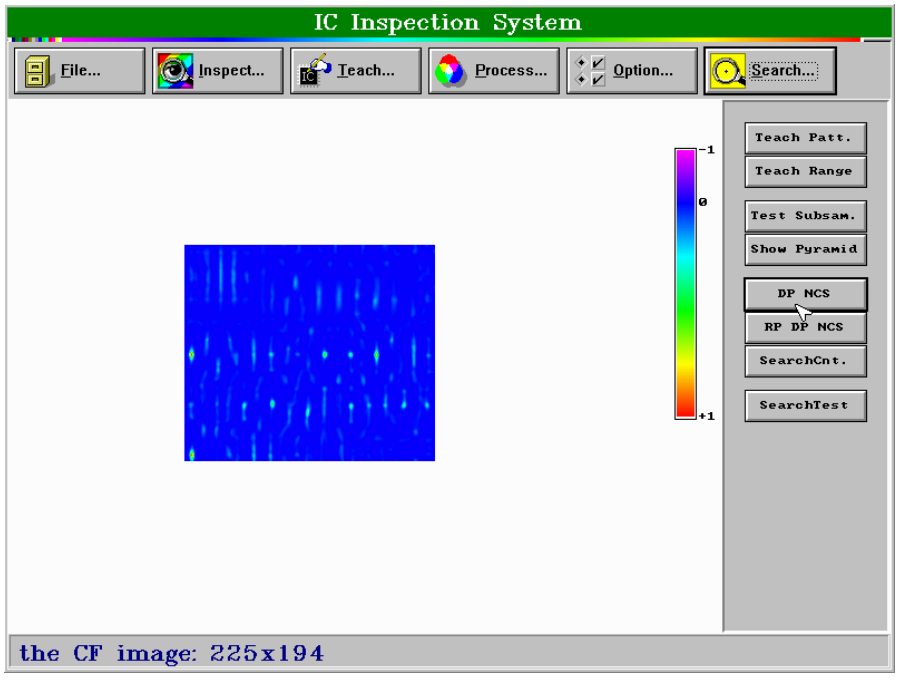

Figure 3.12 : *The correlation function map.*

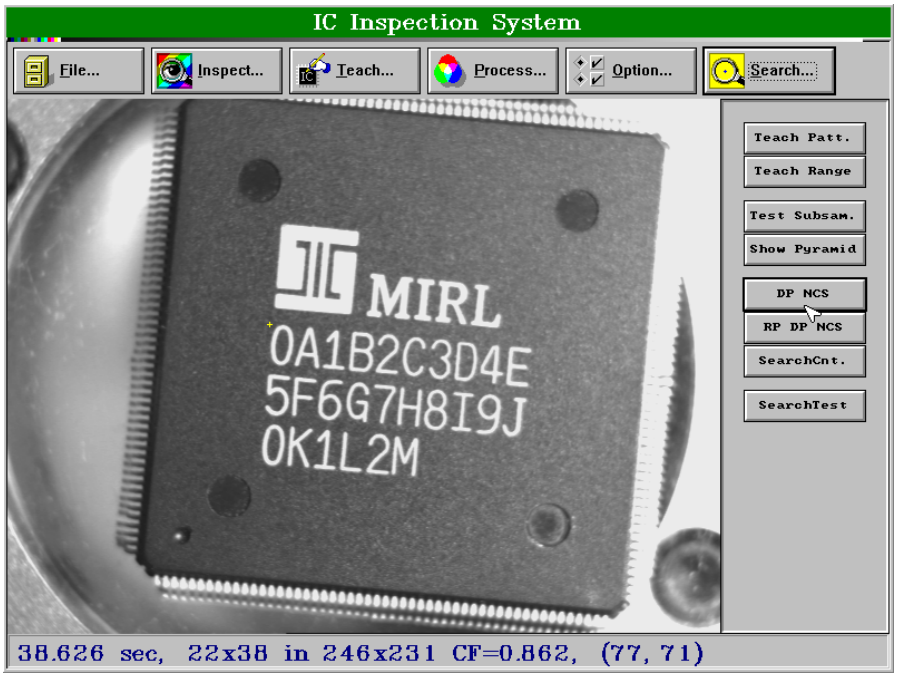

Figure 3.13 : *Test search in a slightly rotated image.*

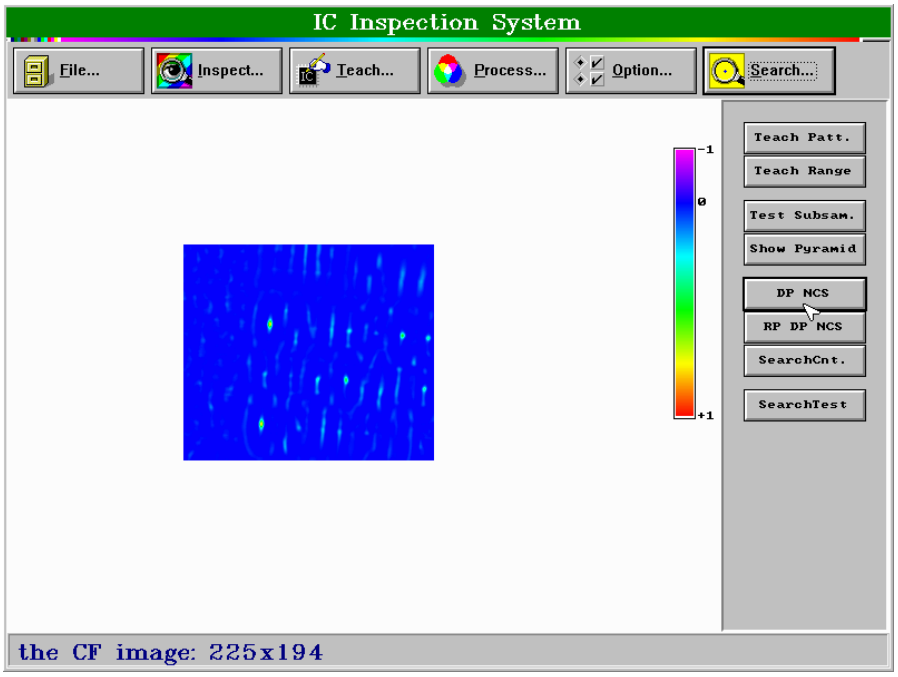

Figure 3.14 : *The correlation function map of a slightly rotated image.*

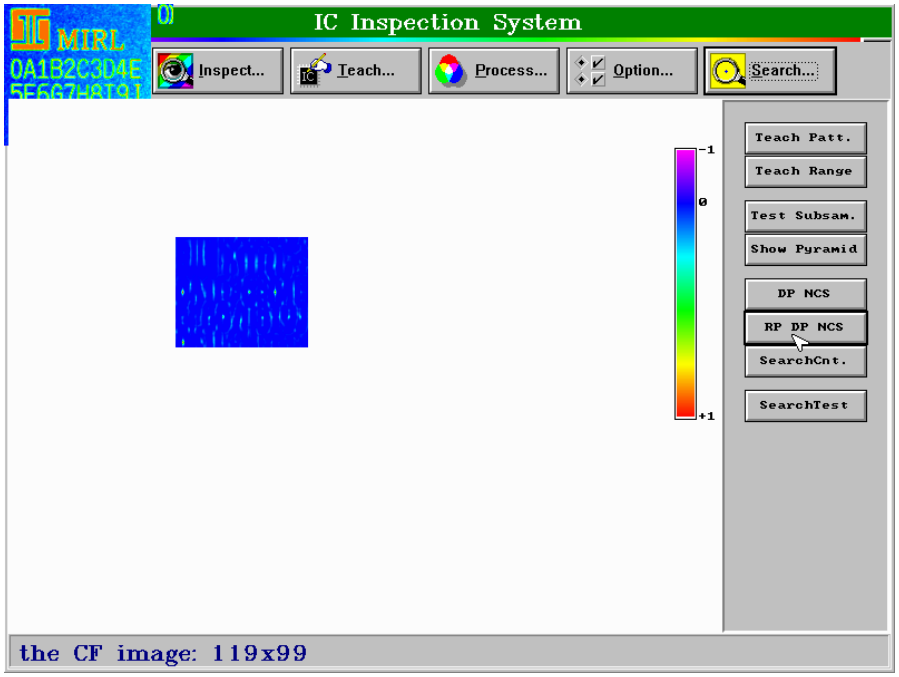

Figure 3.15: *The correlation function map of a sub-sampled image layer.*

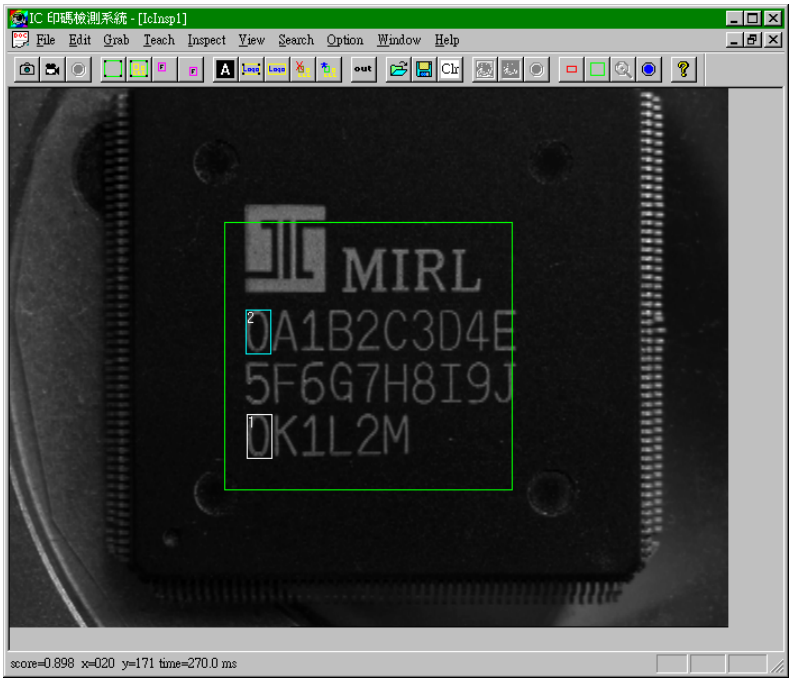

Figure 3.16: *Two targets are successfully searched in an under-lighted image.*

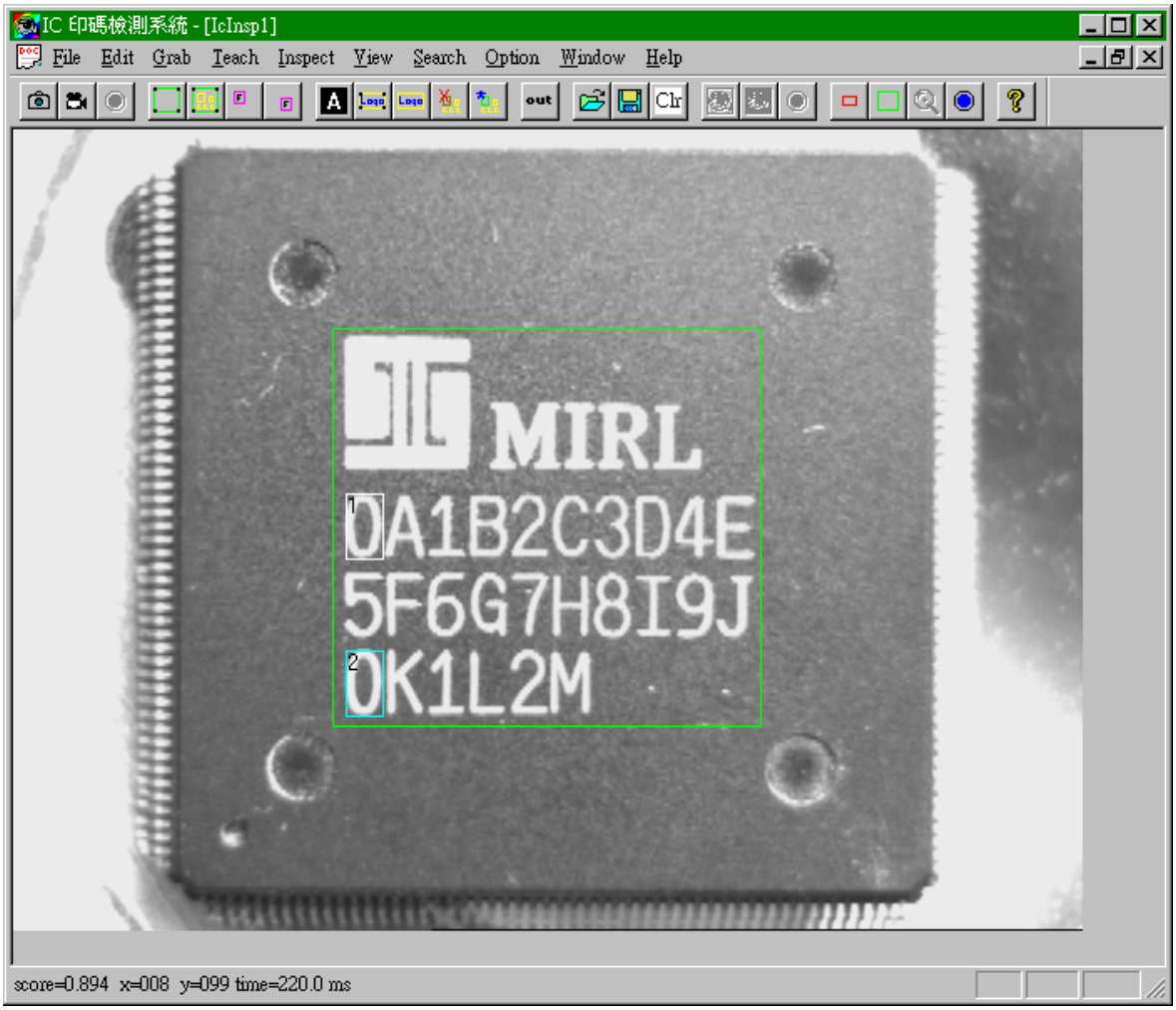

Figure 3.17: *Two targets are successfully searched in an over-lighted image.*

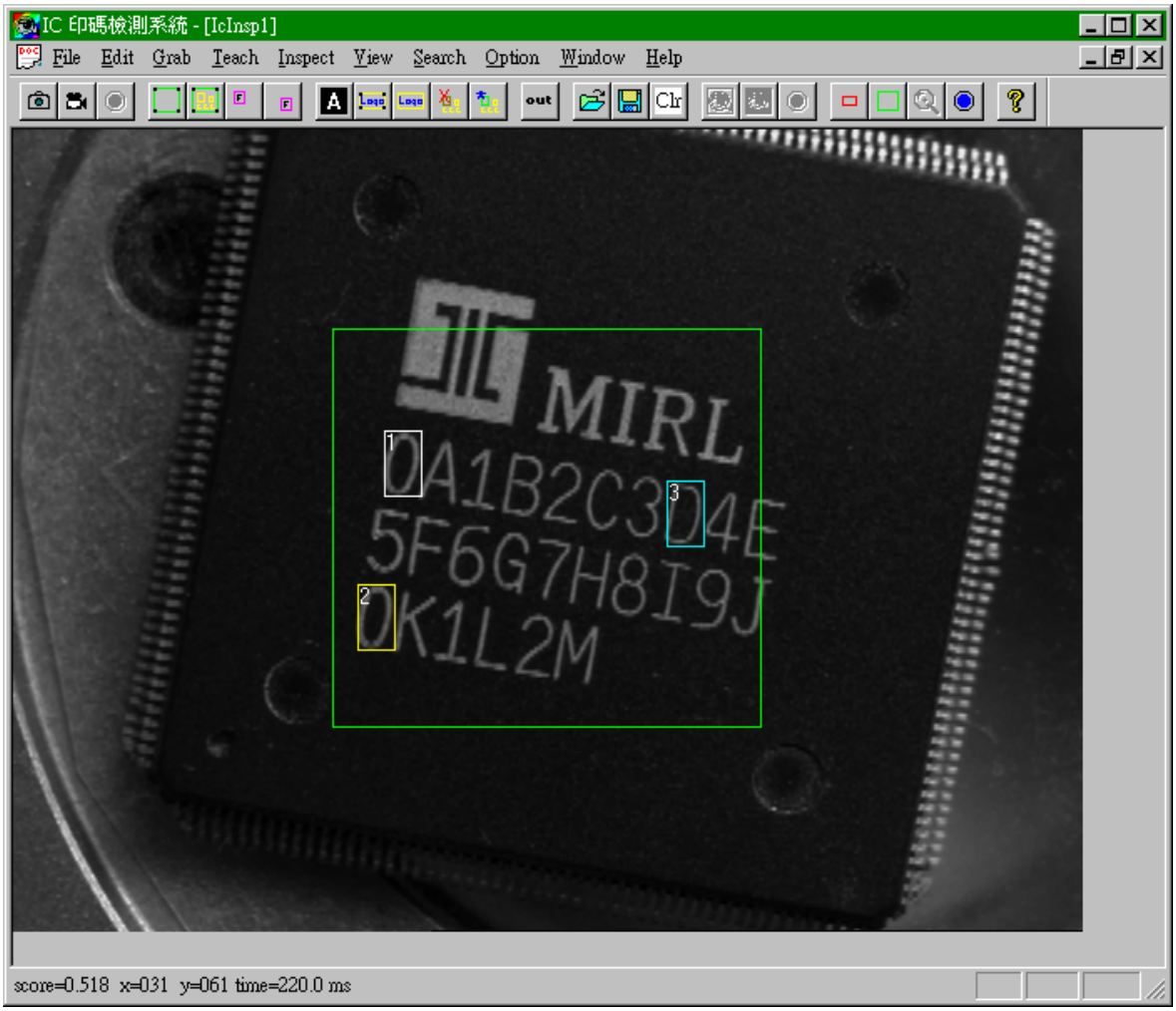

Figure 3.18 : *Two correct targets and one similar target are found in a rotated and under-lighted image.*

Chapter 4

IC Printed Mark Quality Inspection

4.1 Introduction

Integrated Circuits (IC) are the fundamentals of computer and electronic industry. IC industry includes wafer fabrication process in the front end and chip packaging in the back end. IC printed marking is the final stage of chip packaging to print product number and trade mark on the chip to identify product function and classification. IC chips are mass produced. Traditional inspection is manual, dependent on eyes, and prone to mistakes. Automatic IC printed mark quality inspection is a positive and unstoppable trend.

IC printed mark quality inspection is an application of optical verification which requires alignment matching and visual inspection. The IC printed mark includes a logo pattern and characters. Due to the alignment error of the inspection machine, the mark can be rotated or translated. Main printing error of an IC mark includes: distortion, missing ink, wrong position, double print, smear print, bad contrast (global or partial character), misprint, mis-orientation print as shown in Figures 1.1 and 1.2 [2, 6].

The inspection procedure includes the teaching step and the inspection step. We

build a GUI (Graphical User Interface) environment for the operator to teach and adjust a good IC mark sample and then inspect a batch of ICs based on the golden sample.

After teaching, the system will perform character segmentation and feature extraction. Horizontal and vertical projections are used to segment each character and logo. Character or logo after segmentation is called *sub-feature* in our method [7]. Sub-feature is the basic unit of the difference inspection. After inspection, we count the defect proportion of the sub-feature and decide to accept or reject this chip.

We choose two *fiducial marks* from the sub-features as the search pattern for alignment. During inspection, we search for the two fiducial marks to calculate the translation and rotation of the inspected image. Then we rotate back the inspected part of the image and perform pattern difference.

Due to the alignment error of rotation and translation, the segmentation error, and some inevitable noises of CCD camera and frame grabber, edge noise will remain after pattern difference. Morphological opening will eliminate the edge noise and preserve the defect pixels. At last we count defect proportion of each sub-feature and compare it with the acceptance threshold to accept or reject this chip.

Grayscale image is used throughout the inspection process, so appropriate threshold values are important and greatly affect the inspection performance. These system parameters are related to the environment, the defect expected to be detected, the criterion of inspection, and the accuracy of the alignment of ICs. The environment factors include the light source, the IC surface texture and reflection, and the contrast of IC printed marks.

After inspection the IC will be classified into an accepted part or a rejected part. Reliability, repeatability, false alarm rate, and mis-detection rate will be used to adjust the algorithm and parameters. The inspection time is critical and affects industrial implementation. Our proposed method uses the fast search algorithm discussed in

Chapter 3 to save the inspection time and preserve the accuracy.

Sections 4.3 and 4.4 will explain the teaching and inspection process in detail. Section 4.5 will discuss the inspecting parameters. Section 4.6 will show some experimental results.

4.2 The Alignment of IC Image

4.2.1 Two Fiducial Marks to Detect IC Translation and Rotation

We can use two fiducial marks to detect IC rotation as shown in Figure 4.1. Fiducial marks are two characters or logos assigned by user during teaching. By searching fiducial marks in some bounded area, we can locate the test IC and compare the slope angle to the taught data, and then we can calculate the translation and rotation of the test IC image.

Choosing good fiducial marks is important and related to correlation performance. Selecting good fiducial marks can decrease inspection false alarm and increase inspection speed. The selection criteria are described as the following:

- choose the globally or locally distinct fiducial marks, avoiding the ambiguity problem.
- choose the squarer fiducial marks when possible to reduce the processing time by increasing the number of pyramid layers.
- choose the farthest fiducial marks when possible to preserve more precise information for slope angle.

After detecting the rotation angle, we can rotate back the test IC image using the rotation formula:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \tag{4.1}$$

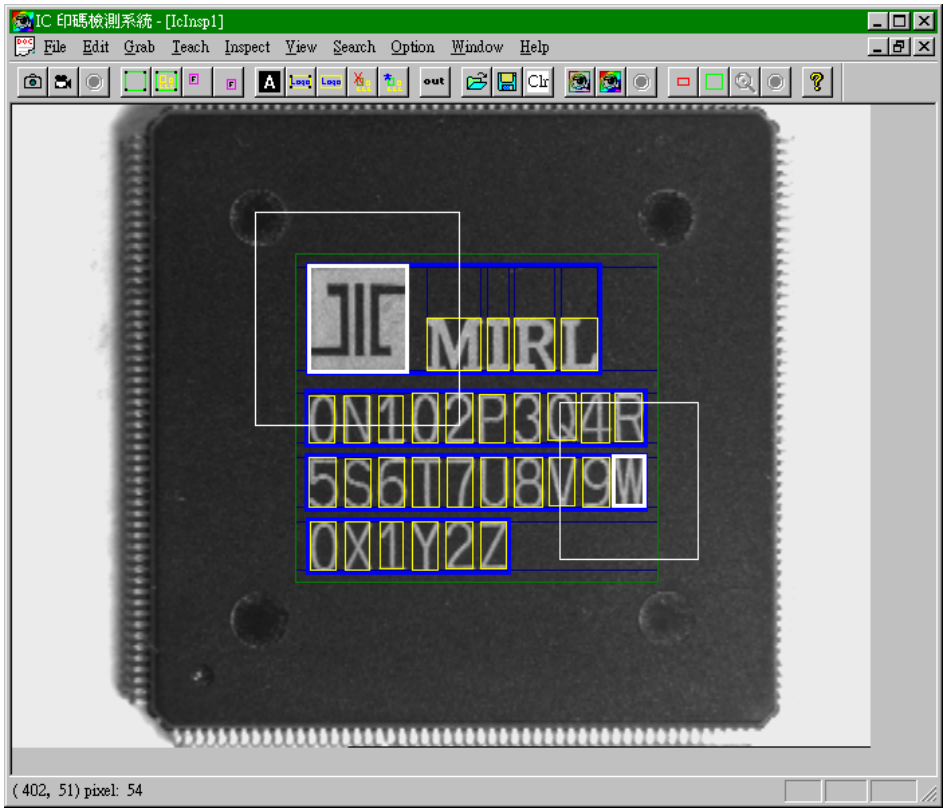

Figure 4.1 : *Two fiducial marks to detect IC translation and rotation. The fiducial marks are the bold rectangles, and the search ranges are the thin rectangles. By calculating the slope angle of the two fiducial marks and comparing with the taught data, we can find the translation and rotation of the tested IC image.*

4.2.2 Rotating Back the Inspected Area of the IC Image

After we calculate the translation and rotation of the test IC image, we can rotate back the test image and translate it and put it into a new continuous 1D memory buffer. For the speed consideration, we need not rotate the whole test image. We first calculate the

translated and rotated four corner points (x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4) from the inspected area $(inspL, inspT)$-$(inspR, inspB)$. Then we calculate the bounding box (r_inspL, r_inspT)-(r_inspR, r_inspB) of the four corner points (x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4). At last we rotate and translate the area (r_inspL, r_inspT)-(r_inspR, r_inspB) and clip the inspected area as shown in Figure 4.2. The area $(inspL, inspT)$-$(inspR, inspB)$ can be used to perform pattern difference inspection directly.

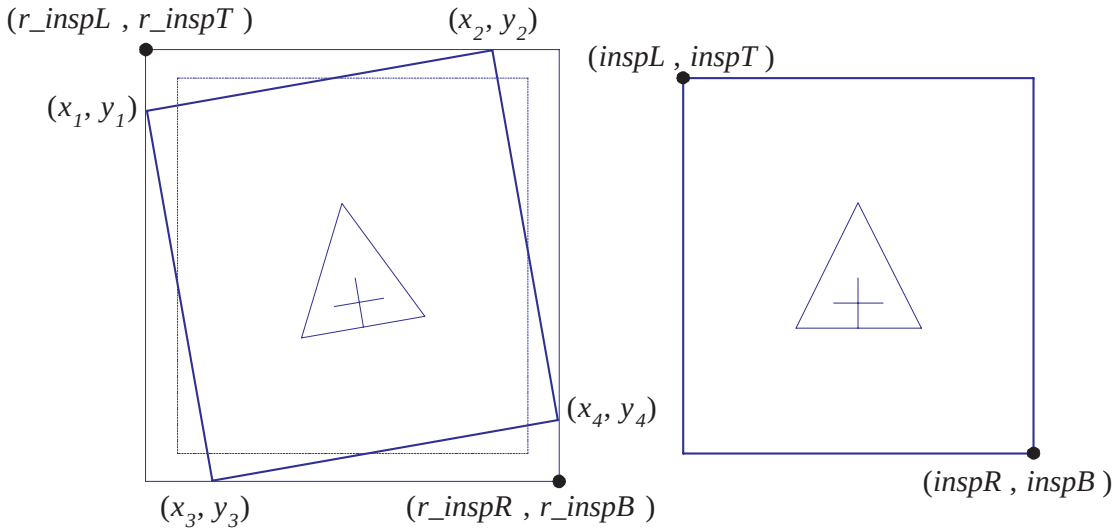

Figure 4.2: *Rotate back the inspected image and compare with the golden IC image.*

4.3 Teaching

The teaching process is shown in Figure 4.3. By the teaching step we can get the golden image and the taught feature data. The inspection parameters will also be set. The teaching is divided into two main steps:

- specify the inspected area and adjust the inspection parameters

- specify two fiducial marks.

After inputting the inspected area, our algorithm will perform automatic thresholding, projection, and noise filtering to segment each sub-feature. The operator will check each sub-feature and the parameters. He or she can adjust the teaching result by the following steps:

- teach logo without projection and noise filtering
- delete improper sub-feature
- adjust the acceptance threshold
- reset the binarizing threshold manually
- segment each sub-feature manually.

The operator can teach multiple inspected areas. At last, two fiducial marks should be carefully assigned or be automatically detected by the procedure described in Section 3.6. The taught data can be saved to disk for later use.

4.4 Inspection

The inspection process is shown in Figure 4.4. Before inspection, the taught data is checked first. The inspection is divided into three main steps:

- search for two fiducial marks
- solve the alignment of translation and rotation
- perform pattern difference, opening, and defect counting.

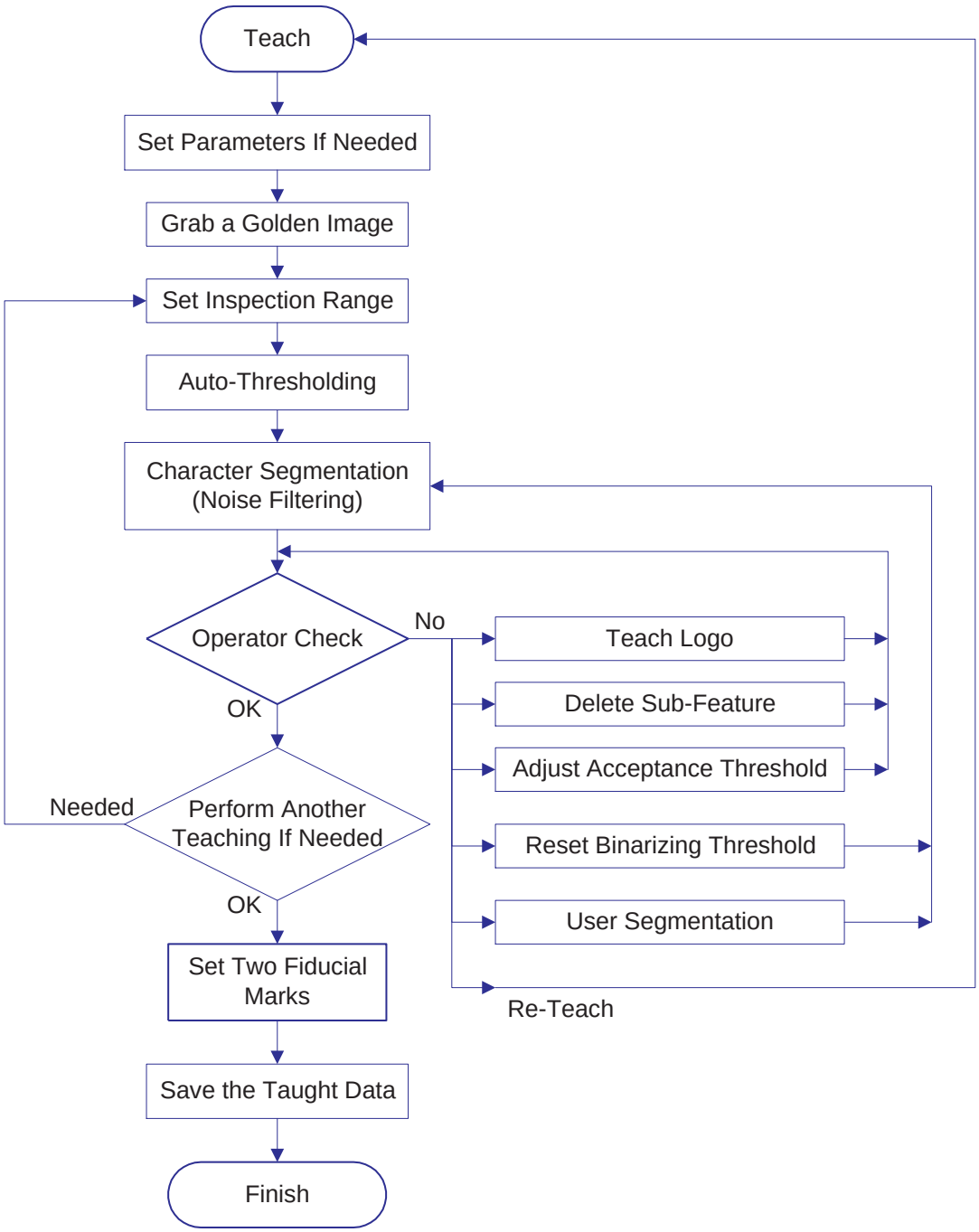

Figure 4.3: *The teaching process.*

	the inspection steps	time (ms)
1.	grab image	50
2.	search for two fiducial marks	50
3.	clip and rotate the image	30
4.	perform pattern difference	10
5.	perform opening	80
6.	count defect, accept/reject	20
	total	240

Table 4.1 : *The approximate time profile of inspection.*

The decision rule to accept or reject a chip is described here. We calculate the ratio of defective part to the foreground part of the sub-feature. The foreground part is the number of pixels whose gray levels larger than the binarizing threshold. If the ratio exceeds the acceptance threshold, we reject this chip. We can adjust the acceptance threshold for each sub-feature as shown in Figure 4.10.

The inspection time is critical for industrial application, but it is related to many conditions such as the hardware, the size of the inspected area, and the inspection parameters. Table 4.1 shows the approximate time profile of our inspection procedure on a personal computer with Pentium 200 MMX.

4.5 Discussion of the Inspection Parameters

The parameters for inspection will affect the performance greatly. To decrease the false alarm and mis-detection rate, we will explain the function of the parameters as follows:

- binarizing threshold:

  grayscale threshold for character segmentation and related to lighting, surface

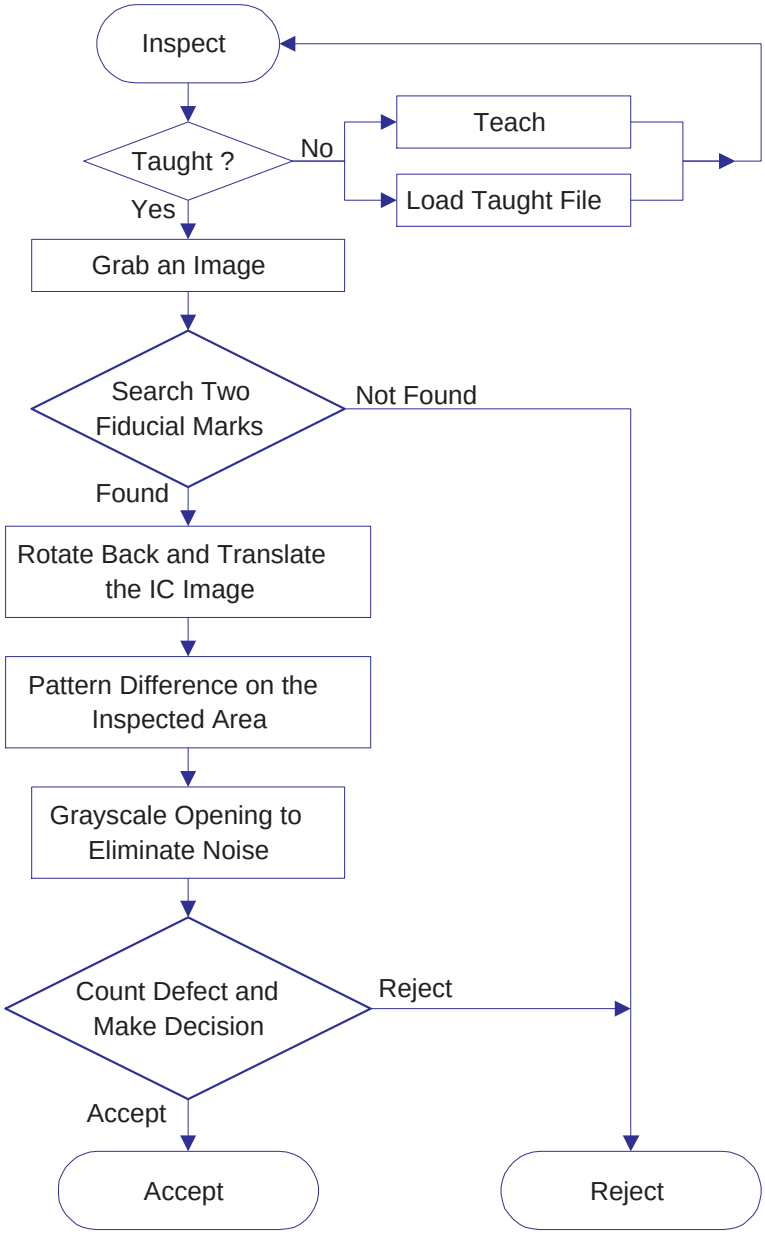

Figure 4.4: *The inspection process.*

reflection, and contrast of printed mark

- difference threshold:
  grayscale threshold after difference inspection and related to the defect to be inspected and the edge noise of alignment error

- acceptance threshold:
  threshold of defect percentage to make decision to accept or reject a part and related to the tolerance and criterion of the system

- other system thresholds:
  such as the correlation matching threshold, the projection thresholds , ... , and so on.

The acceptance threshold can be adjusted for each sub-feature. For example, logo sub-feature with more pixels should be assigned lower acceptance threshold. On the other hand, character sub-feature with fewer pixels should be assigned higher acceptance threshold. Default acceptance threshold is 6 %.

4.6 Experimental Results

Figures 4.5, 4.6, 4.7, and 4.8 show the inspection result of a good IC and a defective IC. Figure 4.9 shows the dialog box to adjust the binarizing threshold during teaching. Figure 4.10 shows the function to adjust the acceptance threshold for a specific sub-feature. We record some images from a real inspection machine in IC packaging fabrication and test them as shown in Figures 4.11, 4.12, 4.13, and 4.14. Figure 4.11 is the good IC image. Figure 4.12 shows the taught data. Figure 4.13 is the defective IC with partial bad contrast. Figure 4.14 is the inspected result that defects are successfully detected. The binarizing threshold here is 115, the difference threshold is 78, and the acceptance threshold is 6 %.

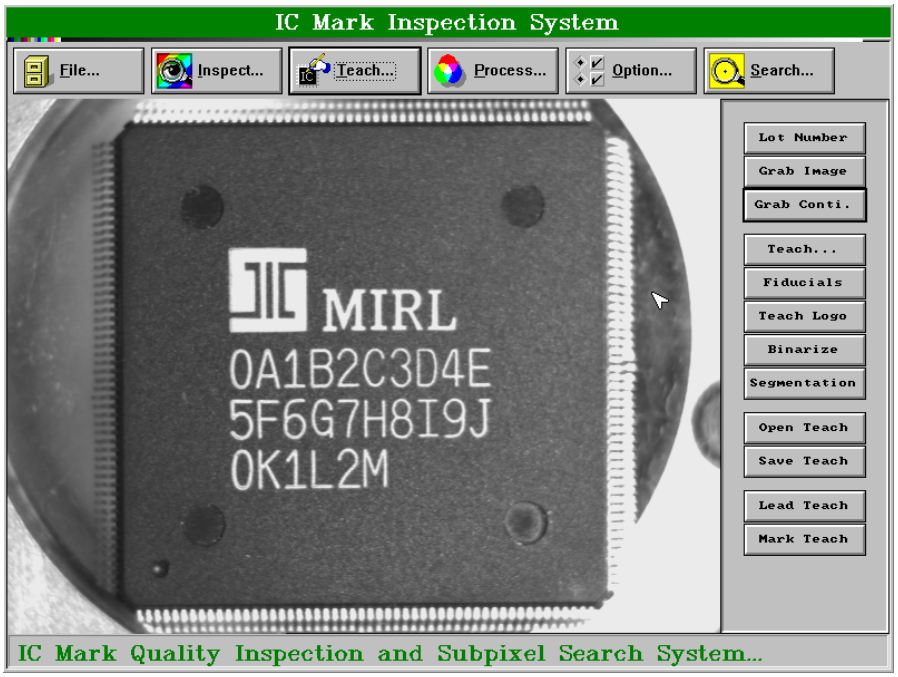

Figure 4.5 : *The good IC image.*

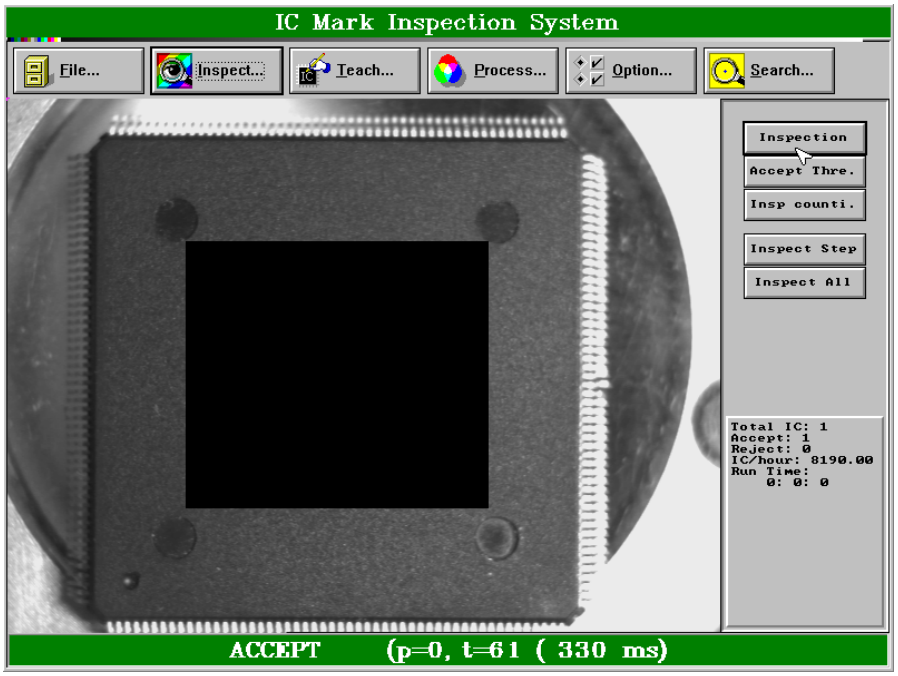

Figure 4.6 : *Inspected result of the good IC.*

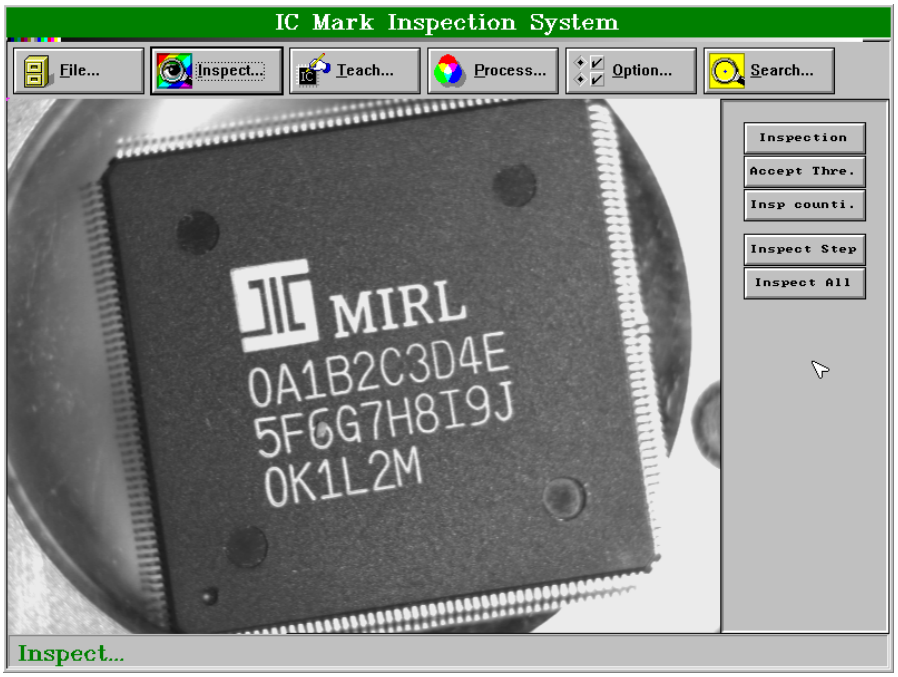

Figure 4.7: *Test image with rotation and defect.*

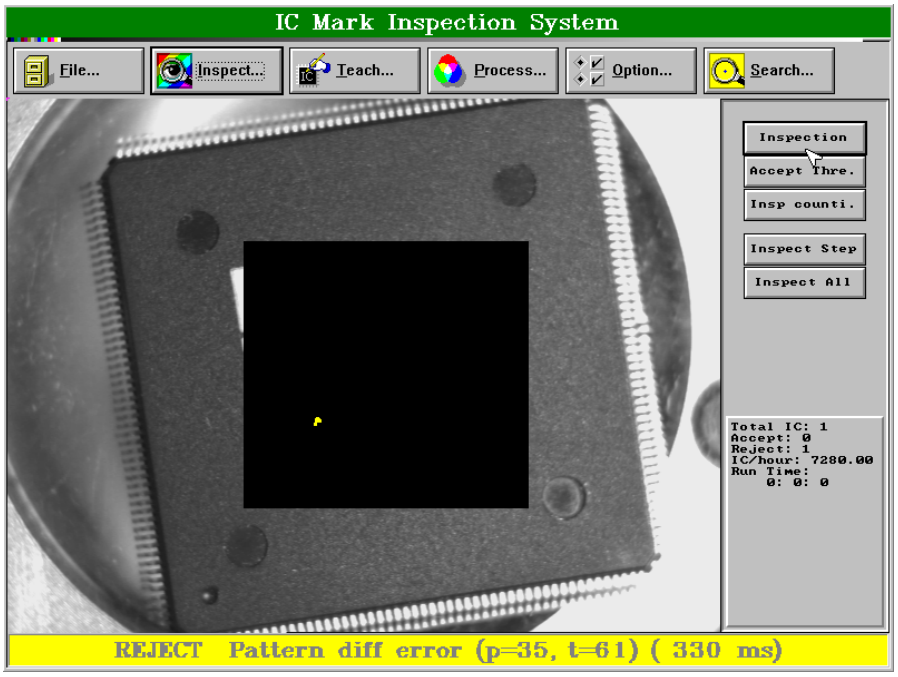

Figure 4.8: *Inspected result of the rotated and defective IC.*

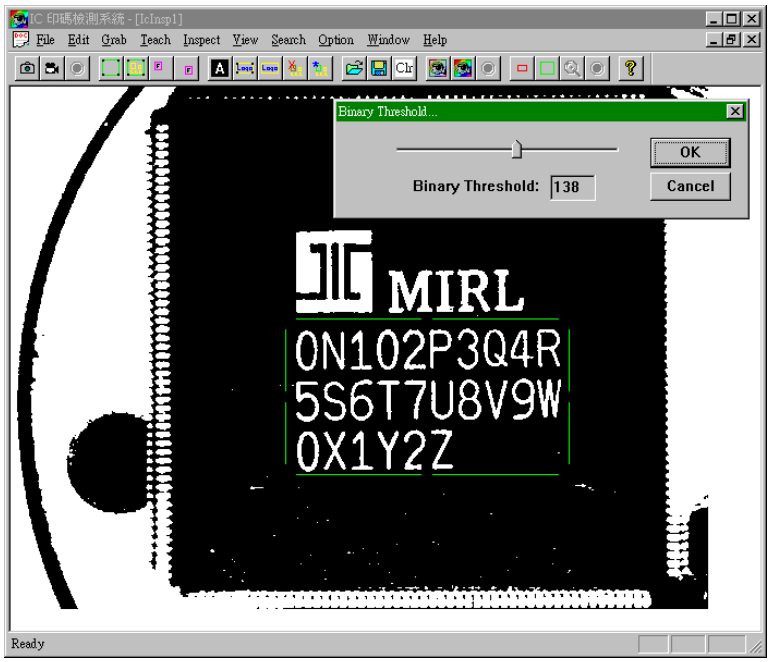

Figure 4.9 : *Adjusting the binarizing threshold.*

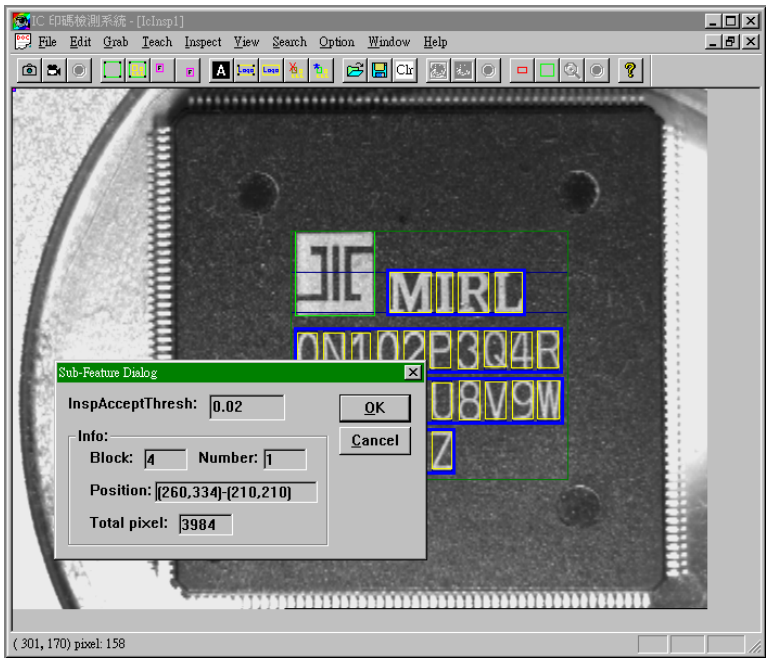

Figure 4.10 : *Adjusting the acceptance threshold for the specific sub-feature.*

Figure 4.11 : *The good IC image.*

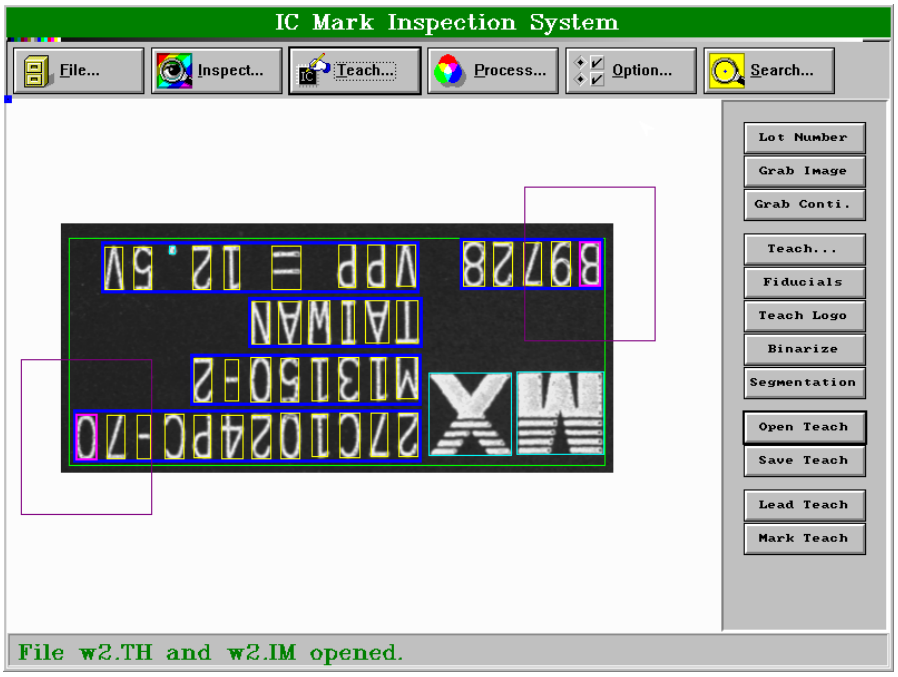

Figure 4.12 : *The taught data from the good IC image.*

Figure 4.13 : *Test image of defective IC.*

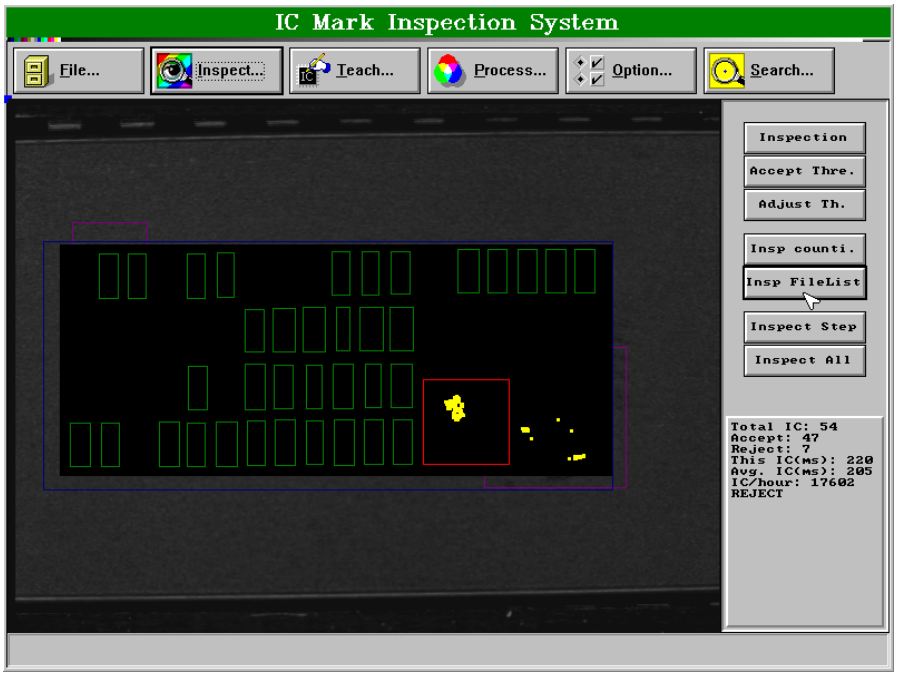

Figure 4.14 : *Inspected result of the defective IC.*

Chapter 5

Conclusion

In this paper, we have proposed a new, efficient, and general purpose grayscale fast search algorithm. Normalized correlation, dynamic programming, and resolution pyramid search are used to achieve the following requirements:

- independence of linear transformation of image intensity
- robustness to slight rotation and image defect
- real-time performance
- subpixel accuracy
- multiple target search
- automatic search model detection.

We also proposed a new and efficient method for IC printed mark quality inspection. The IC may be translated or rotated. We use projection to segment printed characters, use fast search procedure to solve the alignment problem, and use pattern difference and morphological opening to develop the inspection algorithm.

We analyze the functionality and feasibility of each matching algorithms and choose the best trade-off to design our fast search algorithm in a theoretical approach, then we optimize the method by selecting the best search layers and improving the coding technique. As shown in the experimental results, our method achieves high accuracy, reliability, and repeatability with high speed for industrial requirement and works well on field test of various IC products. We have successfully designed and implemented a system for automatic IC printed mark quality inspection.

Bibliography

[1] F. Ahmed, S. C. Gustafson, and M. A. Karim, "Image Interpolation with Adaptive Receptive Field-Based Gaussian Radial Basis Functions," *Microwave and Optical Technology Letters*, vol. 13, pp. 197-202, 1996.

[2] Applied Intelligent Systems, "ValidMark Automated Print Inspection Application Overview," 1990.

[3] D. Bacon, "Image Preprocessing to Improve the Reliability of Normalized Correlation," *Proceedings of SPIE*, vol. 3029, San Jose, CA, pp. 195-199, 1997.

[4] Beck Zaratian, *Microsoft Visual C++ Owner's Manual*, Microsoft Press, Redmond, 1997.

[5] J. Chalidabhongse and C.-C. J. Kuo, "Fast Motion Vector Estimation Using Multiresolution-Spatio-Temporal Correlations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, pp. 477-488, 1997.

[6] M. C. Chang, H. Y. Chen, and C. S. Fuh, "IC Printed Mark Quality Inspection Algorithms," *Proceedings of Conference on Computer Vision, Graphics, Image Processing*, Taichung, Taiwan, pp. 540-547, 1997.

[7] E. A. Chemaly, "Feature Inspection Using Normalized Cross Correlation and Mathematical Morphology," Motorola.

[8] Y. H. Chen, "A Fast Lead Scanning Algorithm for IC Assembly and Leads Inspection," *International Conference on Automation, Robotics, and Computer Vision*, Singapore, pp. 1821-1825, 1994.

[9] H. K. Chung and R. H. Park, "High-Precision Inspection and Alignment of Surface Mounting Devices," *Optical Engineering*, vol. 35, pp. 1820-1827, 1996.

[10] Cognex Corporation, "Auto-Train," *Vision Tools*.

[11] J. Cooper, S. Venkatesh, and L. Kitchen, "Early Jump-Out Corner Detectors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 823-828, 1993.

[12] W. W. Flack, G. E. Flores, and T. Tran, "Application of Pattern Recognition in Mix-and-Match Lithography," *Proceedings of SPIE*, vol. 2440, Santa Clara, CA, pp. 913-927, 1995.

[13] C. P. Gendrich and M. M. Koochesfahani, "A Spatial Correlation Technique for Estimating Velocity Fields Using Molecular Tagging Velocimetry (MTV)," *Experiments in Fluids*, vol. 22, pp. 67-77, 1996.

[14] J. C. Gillette, T. M. Stadtmiller, and R. C. Hardie, "Aliasing Reduction in Staring Infrared Imagers Utilizing Subpixel Techniques," *Optical Engineering*, vol. 34, pp. 3130-3137, 1995.

[15] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, Reading, MA, 1992.

[16] A. I. Hadar and T. Diep, "Lossless Acceleration for Correlation-Based Nearest-Neighbor Pattern Recognition," *Proceedings of International Conference on Pattern Recognition*, Jerusalem, Israel, vol. 2, pp. 240-244, 1994.

[17] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. I, Addison Wesley, Reading, MA, 1992.

[18] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. II, Addison Wesley, Reading, MA, 1993.

[19] M. G. He and A. L. Harvey, "Multi-Directional Diagonal Search Methodology for Object Tracking," *Proceedings of International Conference on Microelectronics and VLSI*, Hong Kong, pp. 347-350, 1995.

[20] A. Hoogs, "Object Position Refinement Using Hierarchical Search," *Proceedings of SPIE*, vol. 2645, Washington, DC, pp. 152-162, 1996.

[21] A. Huertas and G. Medioni, "Detection of Intensity Changes with Subpixel Accuracy Using Laplacian-Gaussian Masks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 651-664, 1986.

[22] J. S. Hwang and C. S. Fuh, "Dynamic Programming to Match Regions in Stereo," *Bulletin of the College of Engineering, N.T.U.*, no. 70, pp. 51-59, 1997.

[23] Imaging Technology, "ITEX/MARK Symbology Inspection Software," 1992.

[24] S. H. Joseph, "Fast Optimal Pose Estimation for Matching in Two Dimensions," *International Conference on Image Processing and Its Applications*, Edinburgh, UK, pp. 355-359, 1995.

[25] A. Kutics, "Evolution of Gray-Scale Morphology Structures for the Extraction of Medical Objects," *Proceedings of SPIE*, vol. 2847, pp. 593-602, 1996.

[26] X. M. Li and C. Gonzales, "A Locally Quadratic Model of the Motion Estimation Error Criterion Function and Its Application to Subpixel Interpolations," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 118-122, 1996.

[27] K. W. Lim and J. B. Ra, "Improved Hierarchical Search Block Matching Algorithm by Using Multiple Motion Vector Candidates," *Electronics Letters*, vol. 33, pp. 1771-1772, 1997.

[28] C. H. Lin, J. L. Wu, and Y. S. Tung, "A Block Matching Algorithm for Near-Real-Time Video Encoding," *IEEE Transactions on Consumer Electronics*, vol. 43, pp. 112-122, 1997.

[29] D. P. Mital and T. E. Khwang, "An Intelligent Vision System for Inspection of Packed ICs," *Proceedings of International Conference - TENCON*, Bombay, India, pp. 1003-1006, 1989.

[30] V. S. Nalwa, *A Guided Tour of Computer Vision*, Addison Wesley, Reading, MA, 1993.

[31] D. W. Paglieroni, G. E. Ford, and Eric M. Tsujimoto, "The Position-Orientation Masking Approach to Parametric Search for Template Matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 740-747, 1994.

[32] M. D. Pritt, "Automated Subpixel Image Registration of Remotely Sensed Imagery," *IBM Journal of Research and Development*, vol. 38, pp. 157-166, 1994.

[33] B. Pudipeddi, A. L. Abbott, P. M. Athanas, and C. C. Weems Jr, "Real-Time Hierarchical Visual Tracking Using a Configurable Computing Machine," *Proceedings of Fourth IEEE International Workshop on Computer Architecture for Machine Perception*, Cambridge, MA, pp. 153-157, 1997.

[34] N. Santosh and C. Eswaran, "Efficient Search Algorithm for Fast Encoding of Image Using Vector Quantization," *Electronics Letters*, vol. 32, pp. 2135-2137, 1996.

[35] M. E. Scaman and L. Economikos, "Computer Vision for Automatic Inspection of Complex Metal Patterns on Multichip Modules (MCM-D)," *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, vol. 18, pp. 675-684, 1995.

[36] M. E. Scaman, L. Economikos, and J. Lambright, "Computer Vision for Automatic Inspection of a High Density Grid of Pads on Multi-Chip Modules (MCM-D)," *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging*, vol. 17, pp. 291-299, 1994.

[37] L. Shen and Y. Sheng, "Noncentral Image Moments for Invariant Pattern Recognition," *Optical Engineering*, vol. 34, pp. 3181-3186, 1995.

[38] W. M. Silver, "Alignment & Gauging Using Normalized Correlation Search," Cognex Corporation.

[39] B. E. Smyth, "Combining Linear and Nonlinear Algorithms to Do Font Quality Verification," *Proceedings of Robots and Vision Conference*, Detroit, Michigan, pp. 13.1-16, 1988.

[40] K. K. Sreenivasan, "Automated Vision System for Inspection for IC Pads and Bonds," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, vol. 16, pp. 333-338, 1993.

[41] M. C. Sullivan and E. J. Wegman, "A Normalized Correlation Estimator for Complex Data Based on a Quadruplex Transformation," *IEEE Signal Processing Letters*, vol. 4, pp. 26-28, 1997.

[42] A. T. Tsao, Y. P. Hung, C. S. Fuh, and H. Y. M. Liao, "On Learning the Threshold Sequence for the Early Jump-Out Template Matching," *Proceedings of AI Workshop*, Taipei, pp. 186-192, 1995.

[43] D. D. Udrea, P. J. Bryaston-Cross, W. K. Lee, and M. Funes-Callanzi, "Two Sub-pixel Processing Algorithms for High Accuracy Particle Centre Estimation in Low Seeding Density Particle Image Velocimetry," *Optics & Laser Technology*, vol. 28, pp. 389-396, 1996.

[44] M. Werman, "Sub-Pixel Bayesian Estimation of Albedo and Height," *International Journal of Computer Vision*, vol. 19, pp. 289-300, 1996.